11-30-2020 11:00 AM

Inverse Mapping of Generative Adversarial Networks

Nicky Bayat, *The University of Western Ontario*

Supervisor: Mohsenzadeh, Yalda, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Computer Science

Abstract

Generative adversarial networks (GANs) synthesize realistic samples (image, audio, video, etc.) from a random latent vector. While many studies have explored various training configurations and architectures for GANs, the problem of inverting a generative model to extract latent vectors of given input images/audio has been inadequately investigated. Although there is exactly one generated output per given random vector, the mapping from an image/audio to its recovered latent vector can have more than one solution. We train a deep residual neural network (ResNet18) architecture to recover a latent vector for a given target that can be used to generate a face image or a spoken digit audio nearly identical to the target. Here we focus on precise latent vector recovery of human face and voice. We use a perceptual loss to embed texture details in the recovered latent vector while maintaining quality using a reconstruction loss. The vast majority of studies on latent vector recovery perform well only on synthesized examples, we argue that our method can be used to determine a mapping between real human faces and latent-space vectors that contain most of the important face style details. In addition, our proposed method projects generated faces to their latent-space with high fidelity and speed. Applying a few further gradient descent steps on predicted latent vectors of face can further improve performance, however the hybrid technique does not help audio inverse mapping. Our audio inverse mapper can reconstruct both synthesized and real spoken digits with high quantitative and qualitative accuracy. At last, we demonstrate the performance of our approach on both real and generated examples.

Keywords: Generative adversarial networks, inverse mapping, latent vector recovery, style transfer, human face images, voice audio

Summary for Lay Audience

Recent Generative models can synthesis a brand new image or audio given a random vector of numbers. The generated output is realistic enough to be indistinguishable from actual real images/audio. Advanced face generators are capable of generating very high quality faces that look as natural-looking as actual human faces[1]. Researchers are improving the architecture and training configurations of these generators everyday. Nevertheless, the task of converting an image or an audio to a vector of numbers, that can reconstruct the target when given to a generator is relatively less investigated. In other work, we know that in order to generate an image, the generator requires a set of input numbers. Given a set of specific numbers, the output of the generator is always the same image. Our goal is to do the backward process. We aim to map an image/audio to its corresponding vector of numbers. Using this predicted vector, one can reconstruct the target image/audio using a generative model.

The world of computer science is about numbers, more specifically 0 and 1s. Finding numerical representations for non-numerical file formats such as image and audio is a common task performed by many different approaches. All the machine learning algorithms are trained and evaluated on the numerical representations of their data sets. Our work is a mapping function that maps human face image and human voice into numerical representations that are in particular useful for Generative Adversarial Networks (GANs).

[1] https://thispersondoesnotexist.com/, visited 2020-11-2

Co-Authorship Statement

The work presented in the second chapter of this thesis was submitted to 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) on October 2020. This paper was submitted in collaboration with Vahid Reza Khazaei and Yalda Mohsenzadeh, with Yalda supervising the project. I am the first author of this paper and I contributed in project conception, method development, experiments and analyses, and writing the manuscript. Vahid contributed in some aspects of coding, and valuable discussions throughout the project, and revising the manuscript. Yalda contributed in project conception, supervising the method development, experiments and analyses, and revised the manuscript.

The work presented in the third chapter was submitted to the 24^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) on October 2020. This research was conducted cooperatively with Andrew Keyes, Vahid Reza Khazaei and Yalda Mohsenzadeh. Andrew and I are both first authors of the paper and Yalda supervised the project. The idea of the project is highly inspired of my work described in the second chapter of this thesis. I have also implemented two alternative approaches to compare with our proposed method and contributed in writing the manuscript. Andrew contributed in conception, implementations, experiments, and writing the manuscript. Vahid contributed in implementing the code, discussions and literature review. Yalda contributed in conception, supervised the method developments, experiments and analyses, and revised the manuscript

Acknowlegements

I cannot begin to express my thanks to Yalda Mohsenzadeh, my supervisor, who has guided me at every step of this thesis. She was there for me day and night, weekday and weekend. She helped me with a wide range of problems, from simple errors in the code to life decisions that came up along the way. Added the Covid-19 pandemic, this masters was no easy job. Her passion for our research kept me motivated and helped me stay on track and succeed.

Many thanks to my lab mates and co-authors Vahid Reza Khazaei and Andrew Keyes who helped me in this research. The completion of my dissertation would not have been possible without your support, friendship and team spirit.

I gratefully acknowledge the assistance of my program coordinator, Janice M Wiersma who helped me during the course of this masters from day one.

Furthermore, I would like to pay my respect to my parents, Mojtaba Bayat and Maryam Mahdinia, who always supported and nurtured me. Their profound belief in my abilities has given me confidence to achieve my goals. Their supportive voice helped me overcome the many ups and downs of this 16 months program.

Finally, to my boy friend, Hisham and my roommate, Reyhanneh who helped me in accomplishing this thesis. You who did not leave me alone during the lockdown, you never stopped believing in me, this would not have been possible without your friendship and emotional support. Thank you so much.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and literature review

1.1 Introduction

Generative adversarial networks (GANs) invented by Goodfellow et al. [16] have numerous applications and are the foundation of many research achievements since 2014. A GAN, which is a class of machine learning frameworks, trains two networks simultaneously, a generator and a discriminator on a given data set. The generator network aims at synthesizing samples as realistic as possible, while the discriminator's objective is to distinguish between real and fake samples. This min-max two-player game motivates both networks to compete with each other and boost their respective performances. Once trained, GANs are capable of generating new samples that have similar characteristics to the training samples. Generators receive random latent vectors that are usually sampled from either uniform or normal (Gaussian) distributions. Lately, progressive growing GANs [23] trained on face benchmarks achieved incredible visual quality in generating fake faces, to the limit that most fake faces are indistinguishable from real human faces. Advanced GANs are also remarkable in generating intelligible audio from a random latent vector.

Most studies use GANs in order to generate desired samples using random input vectors. However, very few works investigated the backwards GAN problem, finding a latent vector that reconstructs an image/audio identical to the input target. Whilst for a given latent vector, there is exactly one generated output, the backward direction might have many solutions. For a given target face, there might be plenty of latent vectors that lead to the generation of a similar face.The same goes for synthesized audio; one fake audio might have many associated latent vectors that are able to generate indistinguishable samples.

Recovered latent vectors can be used as a method to find out about the features a GAN has learned from its training data set; this finding can later be used as a measure of comparing different GAN's performances [5]. In the case that a proper latent vector is not found for an image, we can conclude that certain features in the image cannot be modeled by the generator. This conclusion can be used as a quantitative and qualitative measure to compare performances of different GANs. Another application of latent vector recovery is style transfer; by extracting precise latent vectors of real images, we can transform latent-space vectors of GANs to generate images similar to the natural peers with some added modifications towards a desired direction [53][20]. For example, applying styles to real faces. Style transfer can also be per-

formed in the audio domain; one example of this could be changing the speaker ID of audio. In addition, linear operations on latent vectors result in meaningful changes in generated samples [42]; for instance, adding latent vectors of a person who is smiling to the latent vector of a neutral face will result in a new smiley face. Another example is linearly transforming the latent vector to manipulate the cognitive properties of generated images [15] such as aesthetics, emotional valence and memorability. The inverse mapping of GANs can be useful in a variety of classification or retrieval tasks.

Despite the many applications for inverting generators for real images and audio, researchers have not investigated this issue in much detail. The majority of related publications focus on retrieving latent vectors of synthesized images or audio and fail to reconstruct real examples. Recent latent-vector-recovery methods usually perform poorly on real images or audio, and even those that have demonstrated results on real data sets are rarely tested on the human face and voice. The studies that show results for reconstructing natural faces and audio are tested only on the same data set that GAN was trained on with very high-quality samples. Therefore, we believe there is an urgent need for an approach that is capable of regaining a precise z-space vector for a given natural face or audio, which will be an asset to a variety of applications.

Previous research has established that optimization-based methods can be used to map images to corresponding latent vectors. Creswell et al. [5] proposed an approach to invert any pre-trained GAN using gradient descent optimization with a reconstruction loss. The method starts with a random vector that generates a random image. Next, it computes the loss between the generated face and the target and feeds this loss backwards to update the latent vector. This process is repeated for many iterations until the loss is below a certain threshold. This method recovers the latent vector of an input image in a way, that when fed into the GAN, a similar image to the target is generated. The authors indicate that this inverting GAN technique can be used as both a qualitative and quantitative measure for evaluating GANs. The quality of reconstructed images shows the level and amount of features learned by the GAN, which can be used as a measure of the pre-trained GAN performance. The generated faces used in [5] have very low quality, here we use progressively growing GANs [23] to synthesize more reliable faces with higher resolution. Additionally, [5] evaluate their model only on the same data set with which they trained the GANs, which might not provide information on the method's performance on other natural images. The biggest downside of gradient-based methods is that they tend to fall into local minima, which results in poorly reconstructed latent vectors. On top of that, they are very slow. Stochastic clipping, introduced in [30], can recover the true latent vectors for DCGAN [42] generated images with high fidelity by bounding the recovered vectors to the original training distribution. Nevertheless, these techniques need to converge the gradient descent for each target image separately. For example, [30] needs almost 20,000 iterations of gradient descent to reconstruct good images, and this can be extremely time-consuming, especially when we plan to extract latent vectors of many images. Furthermore, recent GANs are becoming much deeper and are much more challenging to invert than DCGAN [1]. The aforementioned shortcomings have led to the introduction of gradient-free approaches. To the best of our knowledge, there is no previous work that recovers the latent vector of audio using gradient descent. However, we implemented this method to reconstruct audio and compare the result with our proposed method.

Another common approach for extracting latent space vectors of images is training an encoder network alongside the GAN [8][10]. Previous works recovered latent vectors of given

audio through an auto-encoder inspired technique that trains an encoder network either in parallel with the GAN or after the generator is trained [46]. The encoder learns to invert the generated image/audio to its original latent vector. The downside of this technique is that the encoder might overfit to the training samples, which leads to poorly reconstructed counterparts, specifically for those examples that are drawn from a different distribution than the training set. In addition, this method is not applicable to pre-trained GANs because it has to be trained with the GAN at the same time. Moreover, training a third network adds more parameters and therefore is not efficient in resource consumption. The method described in [34] learns an encoder but after training the GAN. Hence, this method is able to handle pre-trained networks. However, the problems of over-fitting and inefficient resource consumption persists. The major advantage of this approach is its high speed. There are solutions to over-fitting problem in Machine Learning, such as cross-validation, regularization, simplifying the model and increasing the size of the training data set. Using these solutions, we can overcome the over-fitting problem and benefit from the high speed of this approach.

Recently, Zho et al. [53] have introduced a hybrid method that benefits from both deep neural networks and gradient-based methods for projecting images into latent-space. First, a similar latent feature vector z to the ground truth latent vector is predicted using a trained deep network with the same architecture as the discriminator of the GAN; next the vector is gradually updated using gradient descent to better match the target image. The reconstruction error used for updating the z vector is pixel-wise Euclidean error. The deep network is trained using a perceptual loss extracted from convolution layers of an AlexNet [28] pre-trained on ImageNet [6]. Predicted feature vector using trained networks is much faster than sole gradient-based techniques, however, it does not reconstruct as visually appealing results. Hence, the latent vector is updated with gradient descent to improve quality. Such hybrid methods are much faster than previous sole gradient-based alternatives. In addition to the high speed, the hybrid method also solves the problem of local minima in the gradient-based method by starting the optimization from a good initial latent vector.

In this thesis, our focus is on the inverse mapping of GANs. We examine the task of recovering the latent vector of both synthesized and real human face and voice. Previous studies have investigated neither natural human faces nor human voice adequately. Additionally, the research to date has not dealt in detail with the inverse mapping of audio samples. While optimization-based methods recover faces with high details, they are incredibly slow and highly sensitive to the resolution of the image. Previously proposed hybrid techniques are faster but have not investigated human face or audio in particular. We believe the human face is a complicated subject that requires more exclusive investigation.

We train a deep residual neural network (ResNet18) [18] to predict latent vectors of given images or audio. We demonstrate our impressive results on generated faces and prove that deep neural networks alone perform magnificently well in projecting generated faces to corresponding latent vectors while being three orders of magnitude faster than optimization-based alternatives. This study focuses on recovering latent vectors of both generated and natural faces and spoken digits. In addition, we extract face details and styles of real human faces in order to generate faces with the same attributes. First, we train ResNet18 on only generated faces using pixel and perceptual loss. Pixel-wise loss is computed between predicted z vectors and their ground truth counterparts. Perceptual loss is computed using the sum of all concatenation layers of a FaceNet [43] pre-trained on MS-Celeb-1M [17]. This approach performs well in

the inverse mapping of generated faces and finds nearly identical recovered latent vectors to the ground truth three orders of magnitude faster than gradient-based alternatives. Meanwhile, it extracts relevant style features of natural faces such as: hair/beard style, facial expressions, pose, gender, etc.

Transferring styles of images using GANs has been the subject of many recent studies. In this investigation, we focus on extracting face details such as hairstyle, gender, pose, facial expressions, etc. while extracting latent vectors of the real faces. Then we use the recovered vector to generate new fake faces that contain similar styles as the real input faces. In order to map natural faces to latent space vectors that contain their style information, we train the network simultaneously on generated and natural face images. We transfer latent-space information from generated faces while extracting detailed facial information using pixel and perceptual loss between the real faces and their reconstructions. Training based on pixel loss between recovered latent vectors of generated images and their ground truth latent vectors (z-loss) helps the model to learn the mapping between image-space and latent-space. The pixel and perceptual loss between reconstructed and natural images aid the network to generate faces with similar attributes to the target.

StyleGAN2 [24] transfers styles between generated faces with high fidelity. Moreover, they generate very high quality synthesized faces. They also reconstruct very high-quality faces based on recovered latent vectors of both natural and generated faces using gradient descent. Nevertheless, their projection to latent-space only works well for very high quality faces; for example, 1024x1024 FFHQ [24] faces, and it is incredibly slow-moving. There are only a few face data sets that include a sample of such high quality, the majority have a much lower resolution. Our proposed method is based on progressive growing GANs [23], which works better for lower quality faces.

With our approach for audio latent vector recovery, we train a deep residual neural network architecture to project audio synthesized by WaveGAN [7] into the corresponding latent space with near-identical reconstruction performance. To accommodate for the lack of an original latent vector for real audio, we optimize the residual network on the perceptual loss between the real audio samples and the reconstructed audio using the predicted latent vectors. In the case of synthesized audio, the Mean Squared Error (MSE) between the ground truth and recovered latent vector is minimized as well. We further investigated the audio reconstruction performance when several gradient optimization steps are applied to the predicted latent vector. Through our deep neural network-based method of training on real and synthesized audio, we are able to predict a latent vector that corresponds to a reasonable reconstruction of real audio. Even though we evaluated our method on WaveGAN, our proposed method is universal and can be applied to any other GANs.

In this work our contributions are the following:

- We propose a fast novel approach for recovering latent vectors of generated faces and audio using deep residual neural networks. This technique reconstructs samples similar to the target much faster than optimization-based solutions.

- Using a combination of pixel, perceptual, and z-loss, we are able to map given natural face images to latent space counterparts that contain similar facial information and style attributes. The reconstructed latent vector can be fed to a GAN to generate an image with identical characteristics to the target.

- We present results of latent vector recovery for images from other data sets (that GAN was not trained on) which proves the generalizability of our model to other data sets.
- We propose a novel perceptual loss within the audio domain using a pre-trained classifier.
- We implement a hybrid method that applies a few steps of gradient descent after predicting the representation by the deep network and compare the results with the proposed Residual Neural Network framework and sole gradient descent approach for both images and audio.

1.2 Related Works

Despite the many applications of GAN inversion, this area is still an open research problem when dealing with natural images, specifically pictures of human faces. Furthermore, this topic has been poorly investigated within the audio domain. Methods focusing on projecting images into the corresponding GAN latent vectors can be categorized into three major groups: encoder-based methods that train an encoder network alongside the GAN or after the generator is trained, for example, methods that train a deep neural network to map images to proper latent-space vectors, optimization-based approaches that start from a random latent vector and update it using a reconstruction loss until it converges and hybrid techniques that first predict a similar latent vector, then they update it to find the best matching feature vector.

Recent advances in the inverse mapping of adversarial generators, in particular inverse mapping of image generators, have led to the question of whether it is possible to map audio to a latent vector that can regenerate it. Despite its many applications, there has been no detailed investigation on recovering the latent vectors of audio. The proposed methods for inverse mapping of images can be applied to audio data as well, but they should be fine-tuned and adjusted for this domain.

1.2.1 Encoder-based methods

Encoder-based methods train an encoder network alongside the GAN [8][10]. GANs are trained to map a random latent vector, sampled from a prior distribution to a more complex distribution (for example images). Such mapping is learned using a combination of a generator and a discriminator. The generators receive a random latent vector as input and project the vector to an image similar to the training distribution. Then the generated image is fed to the discriminator to decide whether it is fake or real. The generator aims to generate realistic samples that can fool the discriminator while the discriminator attempts to find fake samples. The two networks will be trained simultaneously and help each other to boost performance. In order to invert the generator of GANs and map images back into the latent-space, this category of methods trains an encoder alongside the GAN at the same time. So there will be three networks when training: a generator, a discriminator and an encoder. The encoder learns to map the generated image back to its latent vector. [8] calls these networks BiGANs (Bidirectional Generative Adversarial Networks). The architecture of such novel unsupervised feature learning frameworks (BiGAN) is depicted in Figure 1.1.

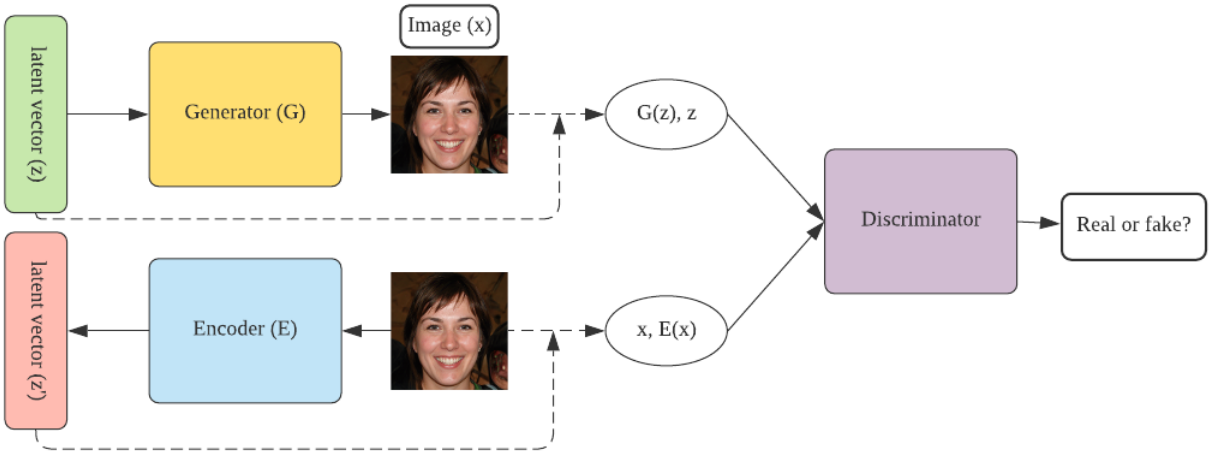

Figure 1.1: The architecture of inverse mapping images back into latent-space vectors via auto-encoder based techniques (also know as bidirectional GANs).

The encoder has no communication with the generator. In other words, E(G(z)) is never computed (E stands for encoder and G stands for generator). However, the encoder must learn the inverse mapping of the generator, through the min-max objective function between the encoder and the decoder, in order to fool the discriminator. The advantage of BiGAN is that it makes no assumption about the type or distribution of the data to which it is applied.

The method proposed in [10] learns both a generator and an inference network through adversarial learning. The generator produces samples using given random latent vectors while the inference network maps images backwards into the latent-space. The discriminator is trained to distinguish between samples from the generator and the images generated using the latent vectors predicted by the inference model. This approach, adversarial learned inference (ALI), treats the inference model as the encoder and the generator as the decoder within an adversarial framework. The encoder and decoder are trained together with the goal of fooling the discriminator. The discriminator distinguishes between real samples of the data distribution and fake samples generated based on the latent distribution.

Previously mentioned methods are either difficult to train or have poor performance. Moreover, the encoder needs to be trained alongside the GAN. Therefore they are not applicable to pre-trained generators. The method proposed in [34] uses the Inverse Generator (IG) model as the encoder. Meanwhile, the authors use the pre-trained generator as the decoder to train the IG model. Their approach, Auto-Encoder based on GANs (AEGAN), tries to minimize the loss between the generated image x and reconstructed image x' rather than the latent vectors themselves. The auto-encoder inspired architecture of this technique is depicted in Figure 1.2.

Finding the inverse mapping through training an encoder network adds more parameters to the training process, which is not efficient in terms of memory usage. Moreover, this network can overfit to the training data, which leads to poor reconstructed latent vectors as well as substandard performance as an evaluation measure for GANs. In addition, such techniques re-

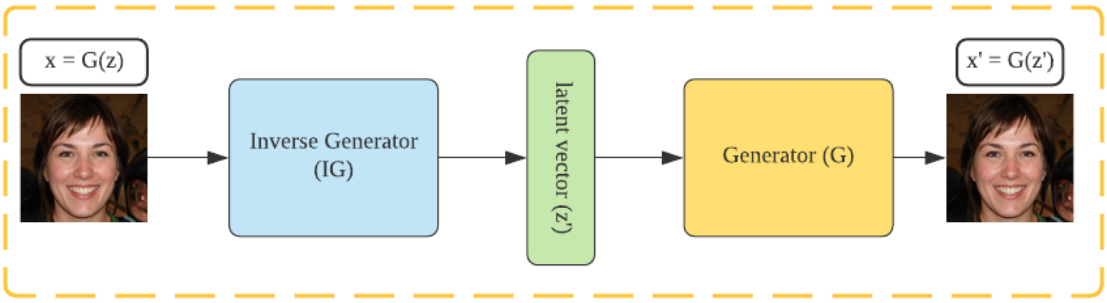

Figure 1.2: The Auto-Encoder inspired framework of AEGAN.

quire simultaneous training with the GAN, meaning they cannot be used on pre-trained GANs. Luo and colleagues [34] have trained the encoder based on a pre-trained GAN. This method overcomes the challenge of handling pre-trained networks.

1.2.2 Optimization-based methods

A large and growing body of the literature has investigated optimization-based solutions [5, 30]. They use gradient descent to optimize the problem of projecting images into the latent vectors that can be used to regenerate those images. They start with a random latent vector, z', sampled from the distribution on which the generator was trained. The goal is to minimize the L2 norm between the target image and the image generated by z' using gradient descent. At each iteration, z' is updated to generate an image more similar to the target until they converge.

Mahendran et al. [35] in 2015 proposed a method that gives insight into the visual information stored in an image by measuring how accurately it can be reconstructed based on its recovered representation. For a target image x, they introduce $\phi(x)$ function that learns a representation for that image. Next, they find the inverse mapping $\phi^{-1}(x)$ that reconstructs image x from its representation. In order to find the inverse mapping, they start from random noise and update it using gradient descent optimization. They use the inversion method to analyze recent CNNs layer by layer. Even though their paper neither is recovering z vectors of GANs nor is focused on face images, their optimization-based approach for inverse mapping $\phi(x)$ is the basis of many gradient-based latent vector recovery studies.

Creswell et al. [5] in 2018 proposed an inverse mapping of GANs, a mapping from data-space back to latent-space. Linearly transforming latent representations leads to meaningful transformations in the resulting images [42]. For example, subtracting a woman without sunglasses from a woman with sunglasses and adding a man to the result leads to generating a man with sunglasses (see Figure 1.5). Therefore, if we find an inverse mapping for GANs, we can use that for classification or retrieval tasks. They used a pre-trained GAN to project images to latent vectors. This inversion technique can give us insights about what features a GAN has learned from the data distribution and use that information to evaluate the GAN's performance.

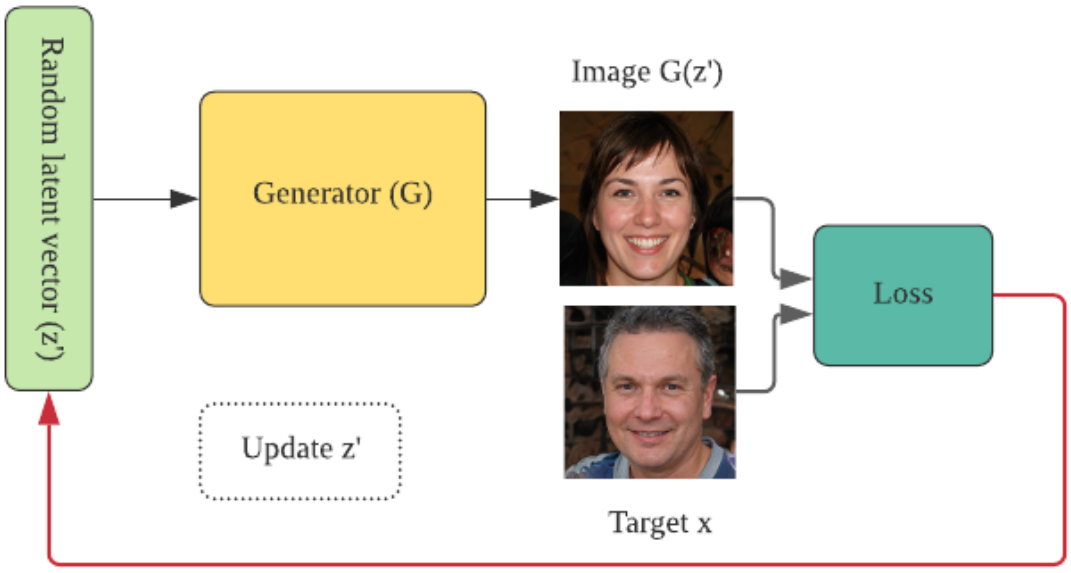

Figure 1.3: The process starts with a random latent vector z' which is updated at each iteration based on the loss between the image generated using z' and the target image until the loss is below a certain threshold. During this procedure, the generator is fixed and the latent vector z' is the only parameter they train.

They use this metric to quantitatively compare different GANs. For a given target image x, they start with a random z^* vector sampled from the distribution the GAN is trained on. The goal is to update z^* in a way that, when passed into the GAN, an image as identical as possible to the target is synthesized. They refer to the projection of image-space to latent-space as inversion. Z^* can be updated using gradient descent methods. They compute the gradient of z^* with respect to G and use this gradient to update z^*. Next, the reconstruction loss between $G(z^*)$ and the target is computed. When the loss is below a certain threshold, the optimal latent vector is found. This minimization process can be formulated using Equation 1.1. The architecture is presented in Figure 1.3. The number of iterations might vary for different target images and some might need up to 100k iterations.

$$z^* = min_z - \mathop{\mathbb{E}}_{x} log[G(z)] \tag{1.1}$$

This approach can be used to invert a batch of images at the same time. They performed experiments on the shoe [50], CelebA [32] and Omnilog [29] data sets. The method presented in [53] computes reconstruction loss based on loss between features of the original generated image and the reconstructed peer extracted from layers of AlexNet [28]. However, the method proposed in Creswell et al. [5] uses a raw pixel loss that is not dependent on any data distri-

bution. In addition, they aim to invert the generator to gain more information about the GAN itself [5]. Hence using an extra pre-trained network might affect the analysis and lead to inaccurately evaluating the GAN. This paper demonstrates results on both fake and real samples that could even come from another distribution than the training data.

Lipton and colleagues in [30] introduced stochastic clipping to bound reconstructed latent vectors to the original distribution the GAN was trained on. Values that are more or less than the allowed range for the latent vector are replaced with a random number with the permitted range. This robust to noise technique performs better than solely gradient descent with exceptionally high recovery accuracy on images synthesized by DCGAN [42], which results in generating images indistinguishable from the target. Different target images require variable number of iterations to reconstruct good images, some might need up to 100,000 iterations in order to obtain a loss below the defined threshold. Thus this method can be extremely slow. Adding small amount of noise to the image will reduce fidelity, nevertheless the reconstructed image still has most of the important features of the target. They minimize L_2 norm to update the latent vector during optimization. Equation 1.2 is the objective function which they aim to minimize.

$$min_{z'} = \|\phi(z) - \phi(z')\|_2^2 \tag{1.2}$$

While the results of optimization-based techniques seem promising, they are usually obtained after numerous iterations of gradient descent, and therefore time to convergence can be a huge barrier when recovering latent vectors of many images. In addition, more recent deep GANs, such as the 15-layer progressive GAN used in this work, are much more challenging to invert than a DCGAN. Additionally, this approach might find the local minimum and finding the global minimum often highly relies on a good starting point.

1.2.3 Hybrid methods

Zhu and colleagues in [53] utilized recovered latent vectors to apply edits to images from the natural image manifold. Editing natural images is extremely challenging since the changes must be applied while keeping the image realistic. This paper mainly focused on editing the shape and color of the images. They claim that there are no highly structured generators that are capable of modeling the exact manifold of natural images. But since GANs has shown promising performance in synthesizing realistic images, they used GANs to generate edited images.

The hybrid method introduced in [53] is depicted in 1.4. For a target input image, they first predict the closest latent vector to the original that produces a similar output as the target. Next, the latent vector z is gradually updated using gradient descent optimizer to be able to generate images closer to the desired target while applying the edits and keeping the image within the natural image manifold. With the assumption that the real image x^R is sampled from the natural image manifold, they aim to find a generated image x^* that minimizes the distance measure $\mathcal{L}(x_1, x_2)$ in Equation 1.3. The distance function $\mathcal{L}$ is the reconstruction error (Equation 1.4) in some feature space C.

$$x^* = argmin\mathcal{L}(G(z), x^R) \tag{1.3}$$

$$\mathcal{L}(x_1, x_2) = \|C(x_1) - C(x_2)\|_2^2 \tag{1.4}$$

Researchers have also proved that using deep neural networks results in generating perceptually meaningful results [9][22]. They used a weighted combination of pixel loss and perceptual loss between features extracted from conv4 layer of a pre-trained AlexNet [28] to recover latent vectors.

After predicting the similar latent vector z^* they optimize it using L-BFGS-B [3] algorithm. While this hybrid method can be very helpful in increasing the quality of the inverse mapping, it relies on a good prediction of the latent vector z. If they start from a random z vector and try to optimize it with L-BFGS-B optimizer, the process is very slow, and it might not converge. In addition, they can start from multiple random z vectors and try to optimize the one with minimum cost. However, the space from which random vectors can be sampled from is very large and choosing the right number of samples to use is not trivial. Instead, they trained a deep neural network to find the initial prediction of the latent vector. Later, they optimized this vector using L-BFGS-B. The hybrid method usually results in better quality compared to sole encoder and higher speed compared to the sole gradient descent method. The objective function of the deep neural network P, with the same architecture as the discriminator of the GAN, is presented in Equation 1.5. x_n is the n-th image of the data set. The Generator is fixed when training the neural network P.

$$\theta_P^* = argmin_{\theta_P} \Sigma_n \mathcal{L}(G(P(x_n^R; \theta_P)), x_n^R) \tag{1.5}$$

The trained deep network finds latent vectors that contain most of the features embedded in the target image. However, the results are not always ideal and can be improved further through a few optimization steps. This hybrid method benefits from the advantages of both neural networks and optimization-based approaches. The generator architecture in [53] is DCGAN [42] that takes 100 dimensional latent vectors as input and generates 64x64x3 images.

1.3 Image style transfer using GANs

Transferring style of a face, for example, the hairstyle, has been of great interest to the research community. Style transfer techniques can be used as a means of editing face photographs that, in particular, is popular in modern social media applications. The human faces can be modified in terms of hairstyle, gender, color of the eyes, age, skin color, pose, shape of the face, etc. With recent advances in deep neural networks and generative adversarial models, applying such edits have become simpler and more accurate.

It has been shown that linear transformation within the latent space results in meaningful changes in the generated image manifold [42]. Additionally, we can update the latent vector linearly by learning a transformation that improves the generated image based on a defined score. For instance, a linear transformer can be trained by feeding the updated latent vector to the generator, receiving the updated image and feeding the image into a classifier that assigns a score to the image in terms of memorability, aesthetics, etc. [15]. This score will be fed

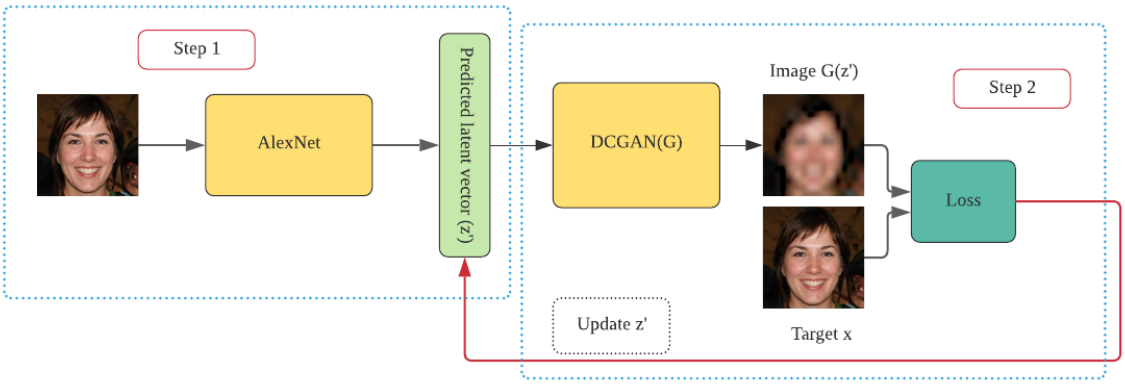

Figure 1.4: The hybrid methodology of recovering latent vectors.

backwards to the transformer in order to learn a better transformation within the latent space. Recent improvements in this area have motivated more researchers to invest in altering the latent vectors to modify the corresponding generated images. Modifying latent vectors to edit images is helpful when editing fake images that have a corresponding z vector. In the case of transferring the style of a natural face, a further step is required to map the real image to a latent vector that can generate an identical face using GANs. Later, this latent vector can be manipulated to apply desired styles to the image. Modifying natural images is much more challenging because the edits must be applied while keeping the image within the natural image manifold.

There used to be two major limitations of GANs. Firstly, GANs were not able to generate high-resolution images due to practical issues; this problem was later solved by introducing progressively growing GANs. Second of all, GANs generate images using random latent vectors. Hence there is no control over the generated faces. Latent vector manipulation has been proposed as a means to alter generated images towards desired outputs.

Radford and colleagues [42] proposed vector arithmetic on face samples in order to evaluate the learned z representations of face images. Mikolov et al. [37] found that simple operations such as the addition of latent representations learned from words lead to rich transformation in the corresponding words. For instance, when the representation corresponding to the word ”Man” is subtracted from the latent representation of the word ”King” and a ”Woman” latent vector is added to the results, the final representation can generate the word ”Queen”.

Arithmetic operations on z vectors of generated images tend to be stable and semantically meaningful when averaged over at least 3 examples of that specific category. Whilst the aforementioned finding is a method of image style transfer using learned representations of GANs, it is itself one of the most common applications of latent vector recovery of images. A sample of addition in z-space is presented in Figure 1.5.

GANs have shown promising results in generating natural-looking images, and linear transformation within the GAN latent-space often leads to meaningful editions in the generated images. Such advantages motivated authors of [53] to first learn a mapping from natural images to

Figure 1.5: Subtracting the latent vector of a woman without sunglasses from a woman with sunglasses and adding the vector of a man without sunglasses will generate a man with sunglasses. The faces used in this figure are generated using a progressive GAN trained on 128x128 CelebA images.

GAN latent space. In other words, they map the image to a feature vector in GAN latent-space that can generate an image as identical as possible to the input. Then they apply user edits on the latent representation. Afterwards, they generate modified images using transformed latent vectors. The latent vector is smoothly updated to apply the user's edit on the generated image while maintaining the image realistic. They edit only the shape and the color of the images. The method proposed in this paper can be a means of generating images based on desired attributes. They use GANs to learn about the space of natural images. This knowledge helps them put constraints on the output of their image editors in order to automatically maintain the edited images as realistic as possible. Their proposed GAN-based method can be used to apply user edits on an image, transform an image to look like another and generate a new image that matches the user's requirements.

The first step is mapping images to the latent space. For that, they first used gradient descent methods. Since deep neural network feature vectors help in perceptually meaningful reconstructions [9][22], they used a combination of pixel loss and perceptual loss extracted from *conv4* layer of AlexNet [28] trained on ImageNet [6]. L-BFGS-B optimizer is used to update the latent vector until convergence, where a good reconstruction depends on a good z vector initialization. In order to solve this problem, they trained a deep neural network to predict a good latent vector to start the optimization with. The deep network P(x;θ_p) directly estimates the corresponding latent vector of a given input image x. The objective function used to train P is presented in Equation 1.5. The architecture of deep network P is the same as the discriminator of the GAN used to model the natural image manifold with slight variations in the number of outputs. After estimating the z vector with deep network P, they optimize it using a few steps of optimization using L-BFGS-B. This method is called a hybrid method of recovering latent vectors of GANs.

When a proper latent vector for a given image is found, then the next step, transforming latent vectors to apply user edits, begins. They smoothly manipulate the feature vector while keeping the image within the natural image space. Each edit is a restriction on a smaller part of

the whole image. The edits mainly include modifications of the shape or color of the objects. The process of applying the edits to the image using GAN latent vectors is summarized in Equation 1.6. They optimize z vector based on the data term, manifold smoothness and E_D term. The data term $\|f_g(G(z)) - v_g\|^2$ makes sure the edit constraints are met, the manifold smoothness term $\lambda_s.\|z - z_0\|^2$ keeps the image inside the natural image criteria and E_D measures the realism of the image generated by GAN from the perspective of the discriminator which enhances the visual quality of the results. They only take 200 gradient descent steps since it takes around 50-100 ms per iteration.

$$z^* = argmin_{(z \in \mathbb{Z})}\{\Sigma_g \|f_g(G(z)) - v_g\|^2 + \lambda_s.\|z - z_0\|^2 + E_D\} \tag{1.6}$$

This method does not modify the original image; it resembles the image with a brand new generated image that later will be modified based on the user's edits. The GAN architecture used in this paper is the same as DCGAN [42] architecture. This approach is limited to the quality of images generated by DCGAN and the information and texture details learned by the generator, as well as the data set the generator is trained on. However, this technique is universal and can be applied to more advanced generators to get better results.

Ganalyze is a novel framework introduced as a measure of modifying cognitive properties such as memorability and aesthetics in GAN generated images [15]. They manipulate GAN latent vectors to generate realistic images with various visual attributes. For instance, the latent space can be navigated towards increased memorability or emotional valence in the output image. The unique advantage of this work is that not only their proposed method is universal and can be used to manipulate latent vectors to generate any desired styles or visual attributes, but also it is a good measure for understanding cognitive features and what it really means for an image to be, for example, more memorable.

Their method starts from a random latent vector z, a random class vector y, a pre-trained generator network G and an assessor function A that measures the score of the generated image based on a given property. They train a linear transformation that, when applied on any random latent vector, can decrease or increase the generated image's memorability (or any other property that the transformer is trained on) by changing the value of a parameter α. The transformation function is shown in Equation 1.7. θ is the parameter that is learned over training. By increasing α, the memorability score increases. The reason they define a class label is that they train their transformer using a pre-trained BigGAN [2], which is conditioned to the input class label.

$$T(z, \alpha) = z + \alpha\theta \tag{1.7}$$

The overall objective function of this transformation is presented in Equation 1.8. If α is equal to zero, it means that the transformation is null, and the image remains unedited.

$$\mathcal{L}(\theta) = E_{z,y,\alpha}[A(G(T_\theta(z, \alpha), y)) - A(G(z, y) + \alpha)^2] \tag{1.8}$$

They trained the transformer based on pre-trained BigGAN [2], and for memorability score, they used MemNet [26] assessor. However, this technique can be used with any generator

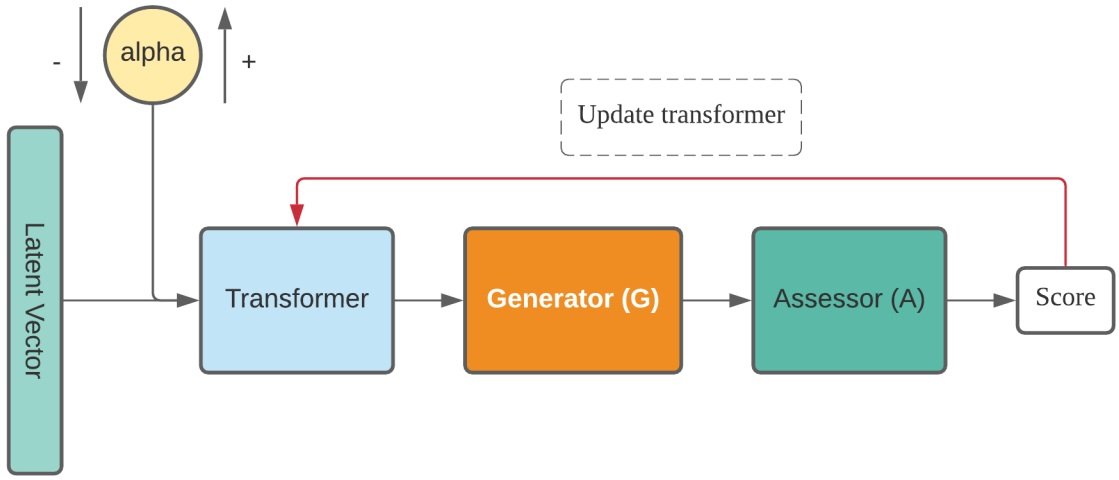

Figure 1.6: The proposed Ganalyze framework that learns a transformer function that is capable of modifying the cognitive properties of the synthesized images.

networks or assessor functions. The overall framework introduced in ganalyze can be viewed in Figure 1.6

With the introduction of progressive growing GANs [23], the quality of generated faces improved significantly. The style-based GAN architecture (StyleGAN) [24] is the most advanced face style transfer tool that utilizes the progressive growing GAN framework and the recovered latent vector idea to transfer the style of faces. Hence a novel opportunity for transferring the styles of both generated and real faces has emerged.

StyleGAN not only generates impressive realistic human faces but also makes changing the style of the generated images possible. Prior to StyleGAN, many works focused on improving the discriminator in GANs to enhance the quality of generated samples. In 2018, StyleGAN paper suggested that we can understand generators better by providing control over the generated images. For this to happen, they introduced a method of disentangling different attributes of the face, which later could be used to manipulate the generator to generate a set of desired attributes. They notably change the architecture of the progressive growing generators. They use bi-linear sampling layers instead of nearest neighbor layers for both up-sampling and down-sampling.

StyleGAN generator does not receive random z vectors as input. Instead, they introduce a mapping network that feeds the styles to the generator as well as noise layers that help in generating even more realistic faces through added stochastic variations. They added the noise to all intermediate feature maps. The style received from the eight fully-connected layers of the mapping network is added to different parts of the generator through adaptive instance normalization layers. They mix the style of two faces by choosing a mixing point. The style of the first face from the beginning to the mixing point will be concatenated with the style of the second face from that point to the end. For instance, in Figure 1.7, the style of the face of

Figure 1.7: Mixing styles of two faces using StyleGAN architecture.

a woman is mixed with the style of another man. The output depicts a mixture of the styles of both images.

StyleGAN2 [25] tackles the problem of existing artifacts in generated faces by StyleGAN by modifying both the training procedure and the generator's architecture. The authors re-designed the normalization method and the progressive growing part. Additionally, they added regularization to the generator to improve mapping from latent vector to images. An example of a face image generated by StyleGAN2 is depicted in Figure 2.4. The path length regularizer also eases the inverse mapping of the generator. They accurately reconstruct both generated and real images using gradient descent. While their reconstructions have very high quality and accuracy, there are two major downsides to their method. First, it takes a long time to recover the latent vector using optimization-based method. Second, they only generate good reconstruction for high-quality 1024x1024 faces.

1.4 Conclusion

In this chapter, the aim was to introduce the inverse mapping of GANs and its most common applications as well as three most common approaches to solve this problem. This chapter has identified the problem of mapping images or audio to latent-space of GANs. The existing literature on inverse mapping of GANs is extensive and focuses particularly on images. Most researchers investigating this problem have utilised optimization-based methods. A number of studies have begun to examine encoder-based approaches to find faster solutions. Some studies attempted to evaluate the impact of hybrid methodologies to combine the advantages of both optimization and encoder based techniques.

A review of recent works that concentrate on editing images using GANs was also presented. This sections is added because later in chapter 2 we demonstrate the ability of our model to embed styles in the generated images by using a real face image as the input to the inverse mapping framework. Collectively, these studies outline a critical role for introducing a fast yet accurate method to project images and audio to corresponding latent vectors.

Chapter 2

Inverse mapping of face GANs

2.1 Overview

In this chapter, our proposed approach for projecting images back to latent space is explained in detail. First, we will emphasize the research problem we are aiming to solve, and then we will explain the method used to tackle the problem. At last, we will describe the experiments we conducted to evaluate the performance of the proposed model. Prior to explaining the method in detail, we will first introduce Generative Adversarial Networks (GANs) and Deep Residual Neural Networks. Later, we will describe the architecture of our model, the objective function used when training and the implementation details. Our inverse mapping model is evaluated on both synthesized and natural face images.

Researchers have recently shown an increased interest in mapping generated samples by generative adversarial networks (GANs) into the original latent vectors. The problem of GAN inversion refers to the task of recovering a latent vector for a given sample (image, audio, video, etc.) that, when given back to the generator, it can precisely regenerate the target. Inverse mapping of GANs has a pivotal role in a wide range of scientific and industrial applications. Despite its many applications, very little has been done in the literature to solve this problem accurately and efficiently.

In this chapter, we aim to find an efficient solution for mapping face images into latent vectors that, when fed into the generator, the synthesized faces are as identical as possible to the target faces.

2.1.1 Generative Adversarial Networks (GANs)

GANs are generative models that are trained to generate realistic fake samples using deep neural networks. In fact, GAN is a training framework for generative models, and it can be used with other models as well; however, [42] showed that more reliable results are achieved via Deep Convolutional Generative Adversarial Networks (DCGAN). This framework is an unsupervised approach that learns the patterns and regularities within the training distribution. After training, the generator is capable of generating samples similar to the training set. The overall architecture of GANs is depicted in Figure 2.1.

GANs [16] train two networks simultaneously. The generator network aims at synthesizing

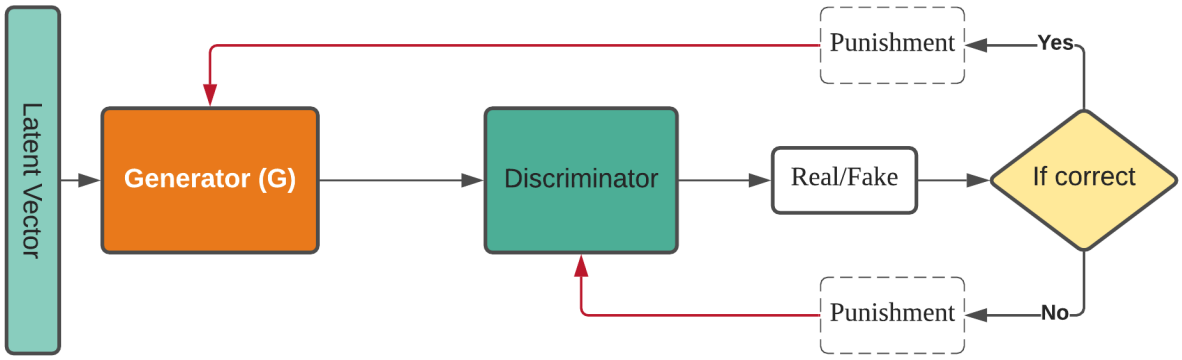

Figure 2.1: The framework of training the generative adversarial networks. If the discriminator guesses correctly, the weights of the generator will be updated else the discriminator will be updated accordingly.

fake samples as realistic as possible, while the discriminator's goal is to distinguish between real and fake samples. This min-max two-player game and the competition between the generator and the discriminator optimizes their performance. When the discriminator detects the sample correctly, it is either rewarded or stays the same. On the other hand, the generator receive a negative feedback loss that is back propagated through its network. Vice versa, when the generator generates a fake sample that fools the discriminator, it either receives a reward or does not change while the discriminator is punished. The training stops when the discriminator is wrong at least half the time. Given a training data set, GANs are capable of generating new samples that have similar characteristics to the training samples. Today, modern GANs are capable of generating samples that even humans cannot categorize as fake.

The generator receives a random latent vector as input. This vector is usually drawn from either a normal or uniform distribution, the same distribution the generator was trained on. Then, the generator generates a fake sample from the problem domain with similar features to the training distribution. The vector space of GANs is referred to as the latent space. This latent space is a high-level description of the generated samples. For each specific latent vector, there is exactly one corresponding generated sample.

The discriminator is a simple binary classifier. It determines whether a given input is real or synthesized. The real data is sampled from the training set, and the synthesized is generated by the generator. The discriminator is only used when training. After that, the generator alone generates synthesized examples.

2.1.2 Fake face generators

DCGAN [42], ProGAN [23] and StyleGAN [25] are three most common face generators available. In this section, these networks are described in more detail and compared against each other. A few synthesized face examples per generator are also provided. In this work, we use ProGAN to train our inverse mapping model. ProGAN generates higher quality images than DCGAN. While the quality of ProGAN generated faces are not as good as StyleGAN, training an inverse mapping system using this generator requires less training time and resources.

Figure 2.2: Examples of fake faces generated by DCGAN model.

Fake face generation using DCGAN

Deep convolutional generative adversarial networks (DCGAN) [42] can be trained on a face data set in order to learn generating fake faces that look as realistic as possible. The difference between DCGANs and original GANs is that DCGANs use deep convolutional neural networks instead of sole fully connected layers used in initial GANs (Vanilla GANs). Convolution layers help finding deep correlation within the image which is in particular helpful when training GANs on images or videos.

When training, both generator and discriminator should have at least one hidden layer. Leaky ReLu works best as the activation function for the hidden units. Since the discriminator works as a binary classifier, the sigmoid activation function is usually used as the last layer of the discriminator. Some examples of faces generated with DCGAN model trained on CelebA face data set [32] (cropped and resized to 64x64x3) is depicted in Figure 2.2.

Fake face generation using progressive growing GANs (ProGAN)

GANs used to be limited by the size of the images they could generate until the introduction of progressive growing GANs [23], a family of generators that incrementally increase the resolution of the generated image until the desired resolution is met. The advantage of this technique is that it can generate higher resolution images than previously 64x64 generated faces by DCGANs. The generator starts generating very small images such as 4x4 then increases the resolution to 8x8, 16x16 and so forth until the image with the requested resolution is generated. The proposed architecture has enabled ProGANs to generate realistic human faces with 1024x1024 resolution.

The novelty of this work is the two-step training process: at every iteration, the first phase includes feeding in a new block of images with higher resolution to the generator, the second phase fine-tunes the generator based on the input given in phase one. Progressive GANs improved quality, stability and variation of the generated images.

When ProGAN wants to increase the resolution, a new block of convolutional layers will be added to both the generator and the discriminator networks. During the fine-tuning step, all layers remain trainable, including the old layer that existed for the previous resolution. This incremental addition of layers helps the model to first learn coarse-level features and later discover fine-level details of the images while slowly evolving.

While ProGAN is capable of generating images up to 1024x1024 pixels, training this network takes a long time. In this thesis, we used a ProGAN trained on CelebA face data set [32] that can generate realistic 128x128 human faces. The framework of progressive growing GANs is presented in Figure 2.3.

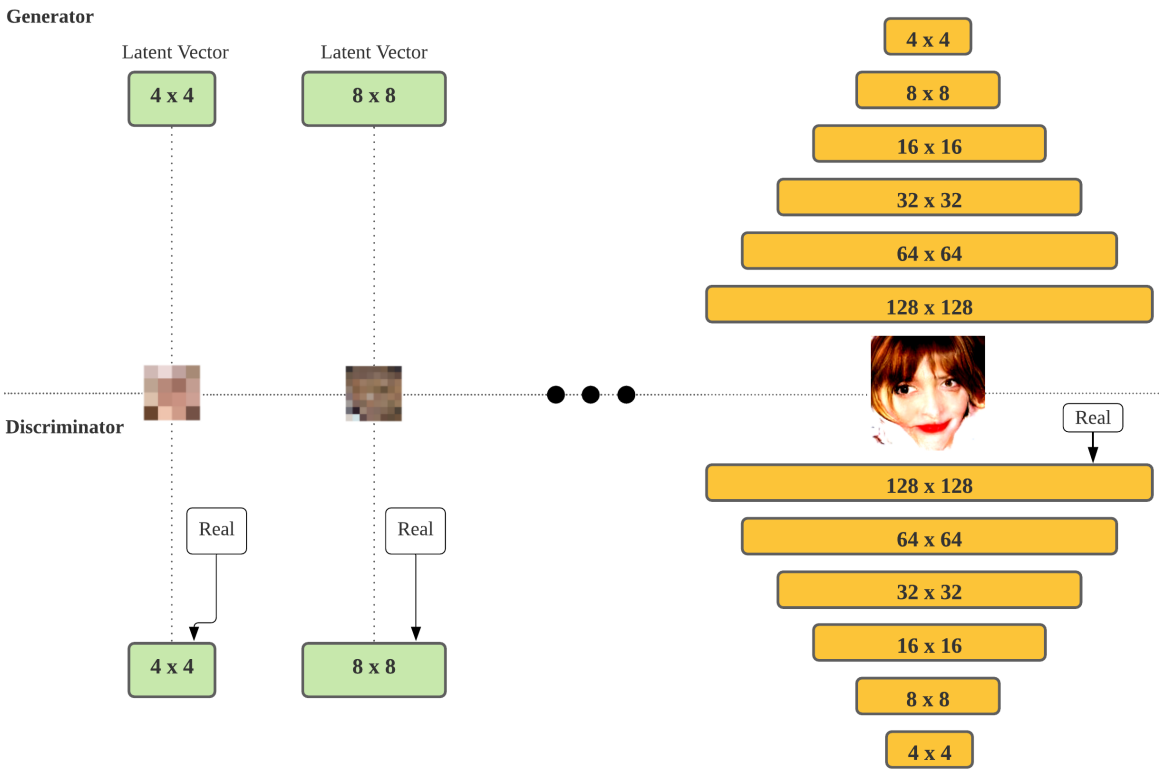

Figure 2.3: The proposed framework in progressive growing GANs.

Fake face generation using StyleGAN2

StyleGAN [25], an open-source project by NVIDIA, aims to generate high-resolution, realistic images and mixing the styles of generated faces. This work modifies the generator in order to edit the style of the image at each convolutional layer. This framework is capable of separating high-level attributes (identity information for example) from low-level details such as hairstyle, gender, etc. which enables StyleGAN to modify some features without affecting others.

StyleGAN generator architecture is proposed to tackle the limitation of GANs in controlling features of the generated images. The additional style information is fed into each convolutional layer that enables users to modify the facial features of synthesized photographs. StyleGAN uses progressive growing GAN architecture to generate high-quality 1024x1024 face images. StyleGAN can generate two images and then generate another image based on these two that take high-level features from one image and the low-level features from the other image.

Coarse-level styles (pose, hairstyle, face shape, etc.) have resolution between 4x4 to 8x8, middle styles are 16x16 to 32x32 and higher resolutions define fine-grained facial styles such as color of the eyes. An example of a face generated by StyleGAN2 architecture is shown in Figure 2.4.

Figure 2.4: An example of a fake face image generated by StyleGAN2 generator.

2.1.3 Face data sets

In this section, commonly used face data sets will be summarized into two categories based on the environment of the face images. Face images are categorized based on whether they are captured under a constrained or an unconstrained setting. Images are mainly pre-processed (aligned and cropped) using multi-task cascaded neural networks (MTCNN) [52] before starting the inverse mapping. Most available face data sets have high resolution face samples that are easier to project into latent-space.

In controlled environments, illumination, pose, facial expressions, occlusion, etc are constrained before taking the photos of the identities. While using such data sets improve results, they are not in accordance with real-life scenarios. The face data set designed and collected by Aleix Martinez and Robert Benavente (AR) [36] is one of the constrained data sets that we

Figure 2.5: Examples of face images from the AR controlled data set.

Figure 2.6: Examples of images from the LFW un-controlled data set.

used for performance evaluation of the latent vector recovery of natural faces. Faces in this data set were taken in two different sessions separated by two weeks. AR contains over 4000 images of 126 identities (70 males and 56 females). Figure 2.5 shows a few samples of the AR data set that are aligned and cropped using MTCNN.

Alternatively, unconstrained environments have no restrictions on the samples, in other words, faces are taken in the wild. Labeled faces in the wild (LFW) [19], VggFace2 [4], CelebA [32], Casia-WebFace [49], MegaFace2 [38] and YouTube-Faces (YTF) [48] are just few examples among many uncontrolled face data sets available.

LFW is a data set specifically designed for the problem of face recognition that includes more than 13000 labeled faces collected from the web. Samples of the LFW data set are presented in Figure 2.6. Vggface2 and MegaFace2 are considered as large-scale face data sets and are mainly used for training. Vggface2 has over 3.3 faces of more than 9000 identities with an average of 362 samples per subject. Casia-WebFace includes 494414 unconstrained images captured from 10575 subjects. CelebA data set is a collection of 202599 celebrity faces with large pose, occlusion, and expression variations that contain 40 attribute annotations and

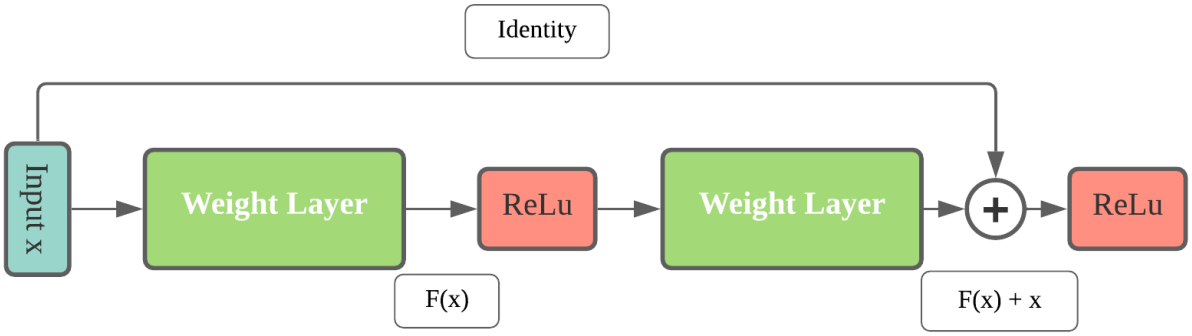

Figure 2.7: A building block of residual learning.

5 landmark locations per image. YTF contains faces collected from 3425 videos that include 1595 subjects and is best suited to the task of video face recognition.

2.1.4 Residual Neural Networks (ResNet)

Deep neural networks, in particular deep convolutional neural networks such as AlexNet [28], have numerous applications and improved the performance of neural networks significantly. Most of the progress of such networks is related to their depth or in other words, their number of hidden layers, which are meant to extract more complex features of the input. Early layers of the network learn lower-level concepts such as edges, mid-layers are more suited to detect mid-level concepts such as shapes and colors and late layers of the deep network are responsible for recognizing high-level details such as object categories.

Unfortunately, there is actually a huge limitation on increasing the number of hidden layers in a neural network. The most common limitation is the problem of vanishing/exploding gradients. The problem of vanishing gradients is caused when computed gradients of activation layers in the deep neural network are very small numbers. Since such derivatives need to be back-propagated through the network to update the weights, they will be multiplied by each other. When very small numbers are multiplied together, the result will be exponentially smaller.

The authors in [18] proposed the residual block to overcome such limitations. Deep networks that contain such residual blocks are called deep residual neural networks (ResNet). Figure 2.7 explains the operation of a residual block via skip connections.

The most important feature of residual blocks is the skip connection. The identity mapping (adding the input of the previous layer to the output of the current layer) has no parameters. If x and F(x) do not have the same dimensions, then a linear mapping W is multiplied by x to match the dimensions. This operation is described in Equation 2.1. It combines the input from the previous layer with the output of the next layer.

$$y = \mathcal{F}(x, W_i) + W_s x \tag{2.1}$$

Skip connections are an incredible addition to DCNNs. They allow training much deeper networks and have empirically shown improvement over the performance of plain convolu-

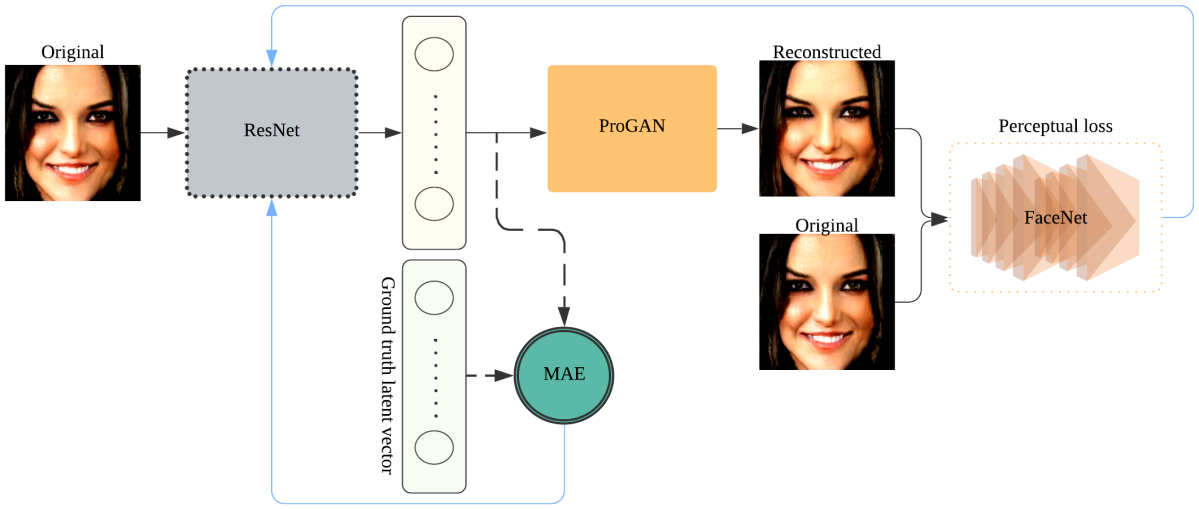

Figure 2.8: The proposed framework for mapping generated faces to latent-space vectors.

tional neural networks on ImageNet classification. Hence we train a ResNet architecture for the inverse mapping task.

2.2 Methodology

In this work, we train a residual neural network (ResNet18) in order to map an input image to its corresponding latent vector using a combination of a reconstruction loss and a perceptual loss. Mean Absolute Error (MAE) is used as the reconstruction loss. The perceptual loss is extracted from the concatenation layers of FaceNet [43] trained on MS-Celeb-1M [17] data set.

We introduce two frameworks: the first architecture trains the network on generated faces, for which we have the ground truth latent vectors. We train the network based on the loss between predicted latent vectors by the network and the ground truth peers (z-loss). The second architecture deals with natural human faces. Since we do not have the true latent vectors of natural faces, we updated the network's weights using two separate objective functions. One epoch, we used a pixel loss and a perceptual loss between the reconstructed face and the target natural face to train the network. The other epoch, the perceptual loss between a randomly generated face and its reconstruction as well as the z-loss between the predicted and true latent vectors, was used to train the model on inverse mapping.

2.2.1 Mapping strategy

Our goal is to train a deep neural network that is capable of mapping samples from image-space into latent vector space (z-space) using a data set of 100,000 generated images and their matching ground truth latent vectors. Since residual blocks are very helpful when it comes to training deep networks [51], we decided to train a residual neural network (ResNet18) to find the mapping. ResNet18 is a residual convolutional neural network architecture that contains 18 layers. The input size of ResNet18 is 224x224; hence we resized all input images to this resolution. We selected ResNet18 over other available pre-trained ResNet architectures because

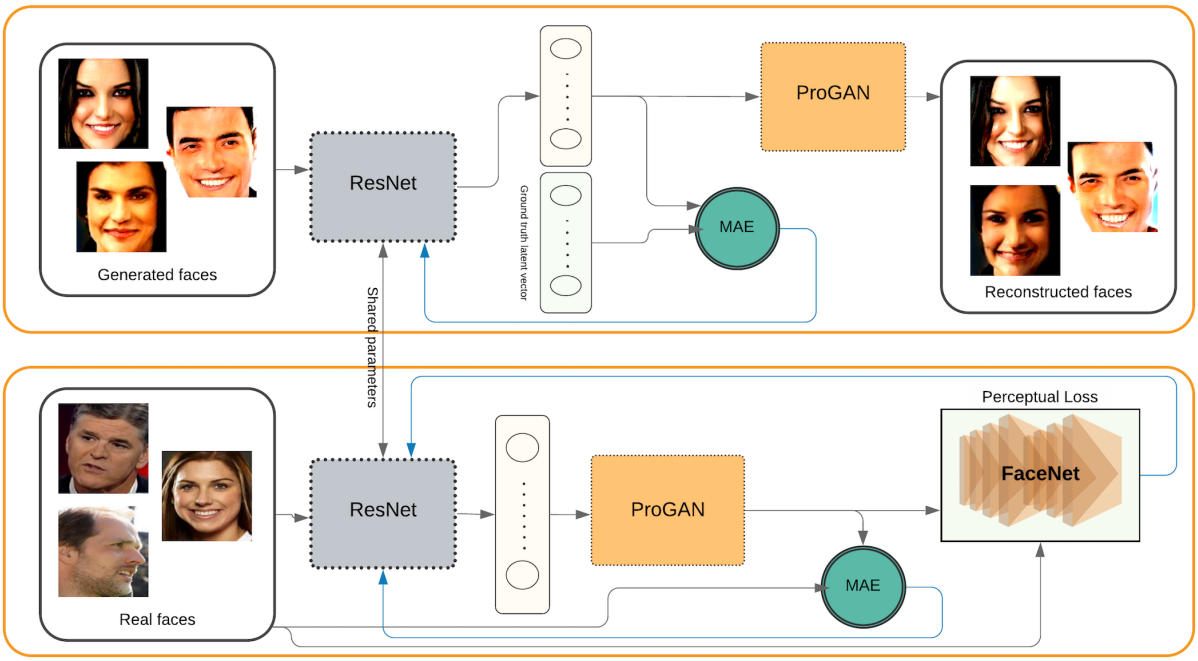

Figure 2.9: The proposed framework for mapping natural faces to latent-space vectors. The generator (ProGAN) is pre-trained on celebA and the weights of this network are frozen when training the inverse mapping model. The same generator is used when training on fake and natural images.

it is smaller (less number of layers means less training parameters); therefore it is easier to train and has a lower chance of over-fitting. A training data set of 100,000 generated faces and their ground truth latent vectors were created using a progressive GAN (proGAN) trained on 128x128 CelebA faces [33] available on Tensorflow hub[1].

When training ResNet to recover latent vectors of synthesized face images, MAE between features extracted from the last layer of ResNet18 (predicted z vector) and corresponding ground truth latent vectors (reconstruction loss), as well as the perceptual loss between reconstructed and target images, were back propagated through the network. Whilst there is only one generated image per given z vector, the inverse problem might have many solutions. In other words, there might be many latent vectors that are able to generate images identical or very similar to the target. Our goal is to find at least one of such vectors and reconstruct the target image based on that. Our proposed method for projecting synthesized faces to latent-space is described in Figure 2.8. Our ResNet18 network trained on a combination of z-loss and perceptual loss is capable of recovering latent vectors similar to the ground truths for given test images with high fidelity and speed. Reconstructed images are almost indistinguishable from the original peers.

Next, we attempted to learn the mapping for natural human faces to latent-space representations. One limitation of natural faces is that they do not have ground truth z vectors assigned to them. In order to overcome this challenge, we trained ResNet18 using MAE and perceptual loss between the reconstructed image and the target. The problem with this approach is that the

[1]https://www.tensorflow.org/hub, visited 2020-11-5

Figure 2.10: The decrease of training loss of the generated images framework over time.

network forgets the mapping from images to GAN latent space. We solved this problem by simultaneous training of ResNet on both natural and generated faces. ResNet18 was trained one epoch using the framework shown in Figure 2.8 and one epoch on pixel and perceptual loss of real faces. The combination of MAE and perceptual loss helped the model to generate images similar to the real target, while training on generated faces with their ground truth latent vectors helped ResNet in learning an accurate inverse mapping. The overview of our proposed framework for real human faces is depicted in Figure 2.9. Our codes and implementation instructions are available on github [2].

2.2.2 Objective functions

In this section, the objective function used when training the ResNet for both synthesized and real faces is explained. Objective function defines a numeric variable loss, which we aim to minimize while training the network. If loss decreases, we know we are moving in the right direction, and the model is learning what it was supposed to learn. The objective function might consist of a combination of different loss terms with different corresponding weights.

Generated faces: When training ResNet on generated faces, we were able to use ground truth latent vectors, random vectors that were originally used to generate training images. The MAE loss between features extracted from the last layer of ResNet (latent vector predicted by ResNet) and the corresponding ground truth latent vector, the z-loss, as well as the perceptual loss between reconstructed face using the predicted vector and the target face (extracted from feature maps of concatenation layers of a pre-trained FaceNet), were fed backwards into the residual network in order to find the best inverse mapping. The decrease in training loss over time can be observed in Figure 2.10.

Natural faces: It is not possible to compute the z-loss for real faces; therefore, the proposed architecture for mapping real faces (see Figure 2.9) is trained one epoch on z-loss and perceptual loss of fake faces and one epoch on pixel and perceptual loss of real faces. The decrease in this loss over time can be viewed in Figure 2.11.

[2]https://github.com/nikiibayat/InverseMappingFaceGANs

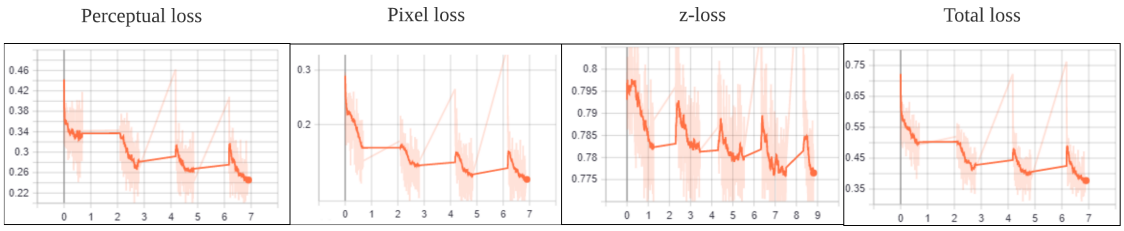

Figure 2.11: The decrease of training loss over time for natural images framework.

Reconstruction loss

The reconstruction loss has the responsibility of penalizing the network for generating images different from the target. This loss can be the Mean Squared Error (MSE) or the Mean Absolute Error (MAE) or the cross entropy between the input and the output. In this work, MAE between latent vectors (z-loss) is used as the reconstruction loss when training on synthesized faces. For real face images, we do not have corresponding ground truth latent vectors; therefore, we use pixel loss between the generated face and the target as the reconstruction loss. The formula for calculating MAE loss between two vectors is described in Equation 2.2 where z^* represents the predicted z vector by our ResNet model and $\mathcal{G}$ is the pre-trained generator, and n is the number of latent vectors in the batch.

$$MAE = \frac{1}{n} \sum \|\mathcal{G}(z) - \mathcal{G}(z^*)\|_1 \tag{2.2}$$

Perceptual loss

Before the introduction of perceptual loss, most models were trained based on a per-pixel loss between the output of the model and the target image. The problem with pixel loss is that it causes blurry outputs. The idea of perceptual loss is to find a measure that captures the perceptual difference between the output of the model and the ground truth peer. This is particularly important in the inverse mapping task because the reconstructed image by the predicted z vector might have perceptual similarity with the target image while having different per-pixel values. For example, two images might be the exact same photo but shifted by one pixel; these photos have a high pixel loss while they are perceptually similar. Perceptual loss is a more robust measure to decide whether two faces are identical or not. This loss also promotes generating higher quality output faces. Perceptual loss compares high-level features such as content and style of the two images. Perceptual loss is not only more reliable but also much faster when optimized.

One common technique to compute perceptual loss is by computing the MSE loss between the features extracted from one or more middle layers of a pre-trained deep neural network on images. The intuition behind this metric is that different layers of a neural network learn different concepts from images. For instance, early layers learn low-level features such as shape and color while middle layers learn mid-level concepts such as eyes. whereas late layers

learn high-level features including a face, a car in a scene and so forth. Based on the task, we can decide the activation of which layer or layers to use to compute the perceptual loss. In most cases, perceptual loss is calculated based on high-level features extracted from late layers of a pre-trained deep neural network. Often times, a combination of perceptual and pixel loss is used in the objective function to obtain desired results.

Recent work on perceptual loss for style transfer [22] shows that it is possible to generate images with high fidelity by defining and optimizing a perceptual loss. The perceptual loss can be extracted from various layers of pre-trained networks. The choice of layers depends on which level of features we want to match between the output and the target. We have tried various combinations of layers to find the best loss function for projecting images to latent vectors. Based on our extensive experiments, we decided to obtain the perceptual loss from all 21 concatenation layers of a FaceNet [43] trained on MS-Celeb-1M data set. FaceNet is a deep convolutional neural network based on the recent Inception networks [21]. It has 426 layers that include a combination of convolution, activation and concatenation layers. FaceNet is trained via an end-to-end approach using triplet loss for the face recognition and verification tasks. This network is capable of mapping each face image into an embedding feature vector in a way that embedding vectors of images with the same identity have less distance compared to those of different individuals. The triplet loss enforces a small distance between faces of the same identity while maximizing the distance between different identities. MS-Celeb-1M is a huge data set of human faces that include more than 10 million images collected from the Internet to support the face recognition task. This data set, published in 2016, is the largest public face recognition data set available. As a result, a FaceNet trained on a huge data set such as MS-Celeb-1M is capable of producing embedding vectors that include important facial details of the input image. Adding perceptual loss to our method objective function results in visually better faces with adequate texture details. The perceptual loss is defined as the sum of the MSE loss between the activation patterns of 21 FaceNet concatenation layers of the target and reconstructed samples. In other words, for each layer, the MSE loss between the activation patterns for both images is computed and added to the total perceptual loss.

2.2.3 Training and implementation details

This work was implemented in Tensorflow 2. Three fully connected layers were added to the ResNet18 architecture with 2048, 1024 and 512 units, respectively. ResNet was trained using an Adam optimizer with a 2e-4 learning rate on a combination of pixel and perceptual loss. The progressive GAN (proGAN), which is used to generate faces, was pre-trained on 128x128 CelebA faces and was downloaded from Tensorflow Hub. A FaceNet trained on the MS-Celeb-1M network was used to compute the perceptual loss.

Data sets: 100,000 faces were generated with proGAN using random latent vectors sampled from normal distribution to create our generated faces training set. In order to obtain our real faces data set, we first selected all the VggFace2 [4] identities that had at least 100 faces, out of which we randomly selected 100 samples per identity and then randomly picked 100 identities. Hence, our final real human face data set consisted of 100 identities, each with 100 example images with a total of 10,000 images.

2.3 Experiments and Results

We conducted five experiments to assess the performance of our face inverse mapping model. The first experiment maps synthesized faces to latent vectors and compares the output with gradient-based techniques both in terms of visual quality and computational time. The second experiment maps natural faces to latent vectors that contain their most important facial feature and evaluates the model in style transfer task. The third experiment utilizes a hybrid method to map natural faces to more accurate latent vectors. The fourth experiment compares encoder-based, optimization-based and hybrid methods based on their performance in the face recognition task. The last experiment evaluates the generalizability of our model to other GANs by mapping fake faces generated by StyleGAN to latent vectors.

2.3.1 Experiment1: Latent vector recovery of generated faces

In order to evaluate the performance of our model in mapping synthesized faces to matching latent vectors, we set up an experiment to both qualitatively and quantitatively compare our results with gradient-based methods.

Gradient-based techniques start with a random z vector drawn from a normal distribution, and then they update that vector using gradient descent for a variable number of iterations. Choosing the right number of iterations and the learning rate is critical in obtaining optimal results. In some cases, updating is stopped when the loss is below a certain threshold, which means a different number of iterations is required for different generated faces, and in some cases up to 100,000 iterations of gradient descent might be required to converge. In this work, we chose 200 iterations to make our comparison viable in terms of computation time.

We generated 50 faces using proGAN to test the recovery of latent vectors using four different methods: gradient descent, gradient descent with stochastic clipping, our ResNet model trained based on only z-loss, and our ResNet trained on both z-loss and perceptual loss. Gradient descent was implemented using an Adam optimizer with 0.01 learning rate and MAE loss between images. Stochastic clipping binds latent vectors to [-1, 1] range, which is the range that z-vectors of proGAN are sampled from during training. If a number is less than -1 or bigger than 1, we simply replace it with a random number within the range. There is another method called standard clipping that replaces any value less than -1 with -1 and any value bigger than 1 with 1. However, Lipton et al. in [30] demonstrated that stochastic clipping is a better clipping technique for this task. Our ResNet18 architecture for this experiment was trained using the framework shown in Figure 2.8. The experiment was conducted on a Nvidia GeForce RTX 2080 TI GPU.

Qualitative results are presented in Figure 2.12. We compared our method trained with solely z-loss and also both z-loss and perceptual loss with gradient-based alternatives with and without stochastic clipping. It can be observed in the figure that adding perceptual loss improves visual quality and reproduces face images indistinguishable from the target (see the comparison of Figure 2.12a (targets) and Figure 2.12e (generated images with recovered latent vectors using our method). Our proposed ResNet-based method is capable of finding a latent representation that can generate a face identical to the target and it is much faster than gradient-based alternatives.

Quantitative results are reported in Table 2.1. We compare the four previously described

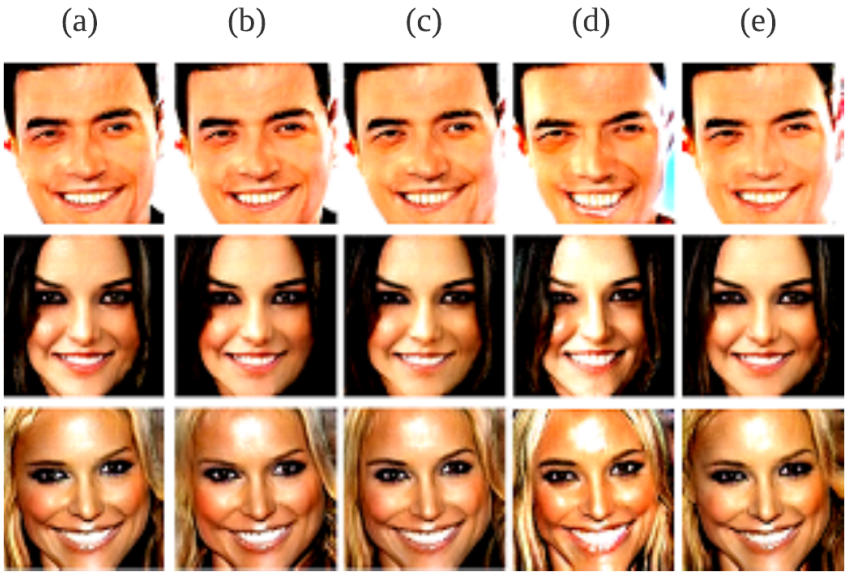

Figure 2.12: Original generated faces are presented in column (a). Column (b) is the generated faces of recovered latent vectors by using gradient-descent method with 200 iterations. Column (c) applies stochastic clipping while updating gradient descent. Column (d) is our method, which utilizes our trained ResNet to map generated images to their corresponding latent vectors. Column (e) is our ResNet trained using both pixel and perceptual loss.

techniques in terms of PSNR metric, Frechet Inception Distance (FID) score, cosine distance and computation time. PSNR is a metric of assessing the quality of a reconstructed image based on the ground truth peer. The formula to compute this metric is presented in Equation 2.3. Cosine distance, which is calculated based on the average of cosine distance between the embeddings extracted from the last layer of the pre-trained FaceNet for both the reconstructed and ground truth target face, is a good measure to compare images in terms of identity information. Our inverse mapping model has lowest cosine distance, we can conclude that our recovered latent vectors reconstruct better facial features and also perform better in identification and verification tasks. This conclusion is further investigated in experiment 4.

$$PSNR = 20log_{10}(\frac{MAX_{original\_image}}{\sqrt{MSE\ between\ images}}) \tag{2.3}$$

What stands out in the table is that our method is significantly faster than other state-of-the-art techniques while having slightly lower PSNR. We compromise visual quality (to a negligible degree) for three orders of magnitude faster latent vector recovery. The difference in PSNR metric is negligible compared to the significantly lower computation time. In terms of the FID score, our method performs slightly worse than gradient-based alternatives. But again, the difference is negligible compared to how fast the our inverse mapping performs.

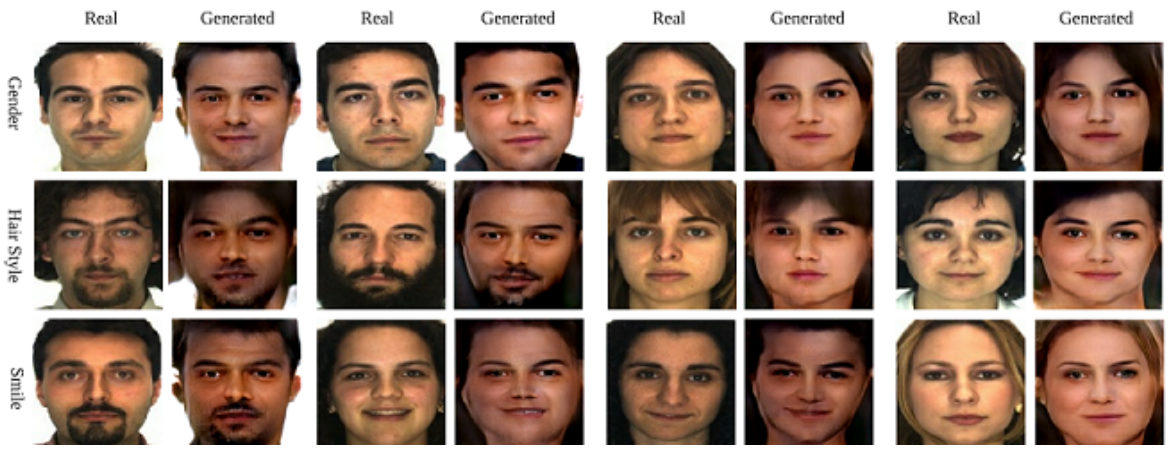

Figure 2.13: The results for mapping natural faces to latent-space vectors that contain same style and facial features. The faces in each row represent how recovered latent vectors understand the gender, hair style and emotions of the target image respectively.

2.3.2 Experiment2: Style transfer using natural faces

Mapping real human faces to latent-space vectors, which leads to regenerating those faces using GANs, is extremely challenging and very little was found in the literature on this topic. Our proposed architecture for natural faces extracts latent vectors that contain important facial information on the given real face. These facial details include face shape, gender, hair/beard style, smile, pose, etc. and will then be applied to the generated faces. This approach can be a means to control the face image synthesis process by altering the latent vector based on given styles.

In order to evaluate our ResNet model in style transformation from real to generated faces, we conducted an experiment on the AR face data set [36]. AR is a strictly constrained face data set that includes over 4000 frontal view face images of 126 identities (70 males and 56 females). Facial impressions, illumination and occlusion of the participants are controlled, but there are no restrictions on what people are wearing (make up, glasses, hair style, etc.). The pictures were taken in the same conditions in two different sessions separated by two weeks. We trained ResNet on VggFace2 data set of real faces and evaluated the performance on the

Table 2.1: Comparing gradient-based methods with ours based on PSNR, FID score, cosine distance and the computation time (in seconds) for each method.

Model	**Metric**			
	PSNR	*FID*	*Cosine distance*	*Time*
Adam	18.41	53.01	1.96	1143.90
Stochastic Clipping	**19.06**	**30.48**	1.98	1135.24
Pixel-ResNet(ours)	11.06	105.85	**1.94**	**3.27**
Pixel-Perceptual ResNet(ours)	16.30	31.22	1.98	3.96

Figure 2.14: Recovered latent vectors preserve the pose of the target face.

AR data set. Please note that the proposed model was not trained on AR data set, and we used this data set for testing.

The findings of this experiment indicate that projecting a real face image into latent-space using a deep residual network trained on pixel, perceptual and z-loss results in a latent vector that can be used to generate a face with similar styles as the input. Figure 2.13 illustrates the style transformation of a generated face given natural input faces on the AR face data set. The recovered latent vector includes information about the gender, hair/beard style, facial expressions (for example smile) and many other attributes of the natural face.

In addition, we conducted another experiment on high-quality pictures of real human faces with different poses to indicate that recovered latent vectors not only preserve hairstyle, gender and facial expressions but also generate faces with the same pose (Figure 2.14).

2.3.3 Experiment3: Hybrid reconstruction of real faces

As previously explained in the literature review chapter, there are three common approaches to latent vector recovery, gradient-based, encoder-based and hybrid methods. A hybrid method first predicts an initial latent vector using a pre-trained model, namely an encoder, next optimizes that vector to improve the quality and accuracy of the results. The Hybrid method improves sole encoder-based techniques because of the additional gradient descent steps. Updating a latent vector using gradient descent slightly changes the latent vector to motivate generating faces that look more similar to the target.

Optimization-based methods require a great number of gradient descent iterations to perform well. Some images might require over 100,000 iterations to be reconstructed with high details. This process is incredibly time-consuming, which makes this method impractical, in particular, when we have many images to map. The advantage of hybrid methods over sole gradient descent is that they start from a good initial point. Thus they need fewer iterations to converge.

Even though our ResNet18 encoder is able to predict latent vectors that contain important facial details of the target, there is still room for improvement. We assumed that such latent vectors could be a good initial point for a hybrid gradient descent optimization. There is a trade-off between the number of iterations and computation time. If we update the latent vector for more iterations, the output looks more similar to the target, but it takes more time to recover the optimal latent vector. If we update for less number of iterations, the overall method is faster but the generated output is not identical to the target.

We experimented with two different values for the number of iterations. The results of this comparison are presented in Figure 2.15. Column (a) are the original AR faces that are used as

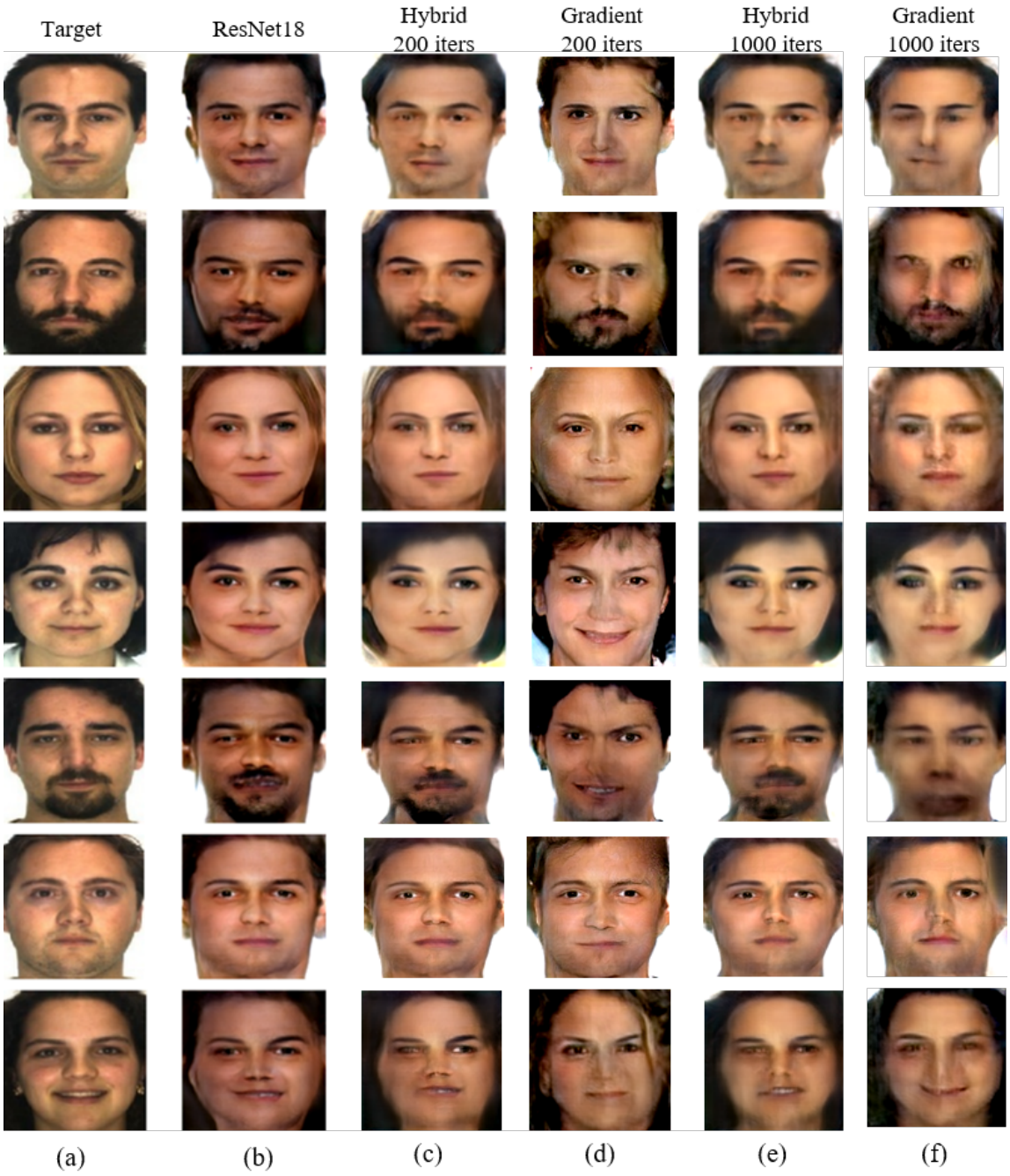

Figure 2.15: The comparison between inverse mapping using a ResNet18 encoder (b), sole gradient descent (e) and a hybrid technique (columns (c) with 200 iterations and column (d) with 1000 iterations) on the natural AR faces data set.

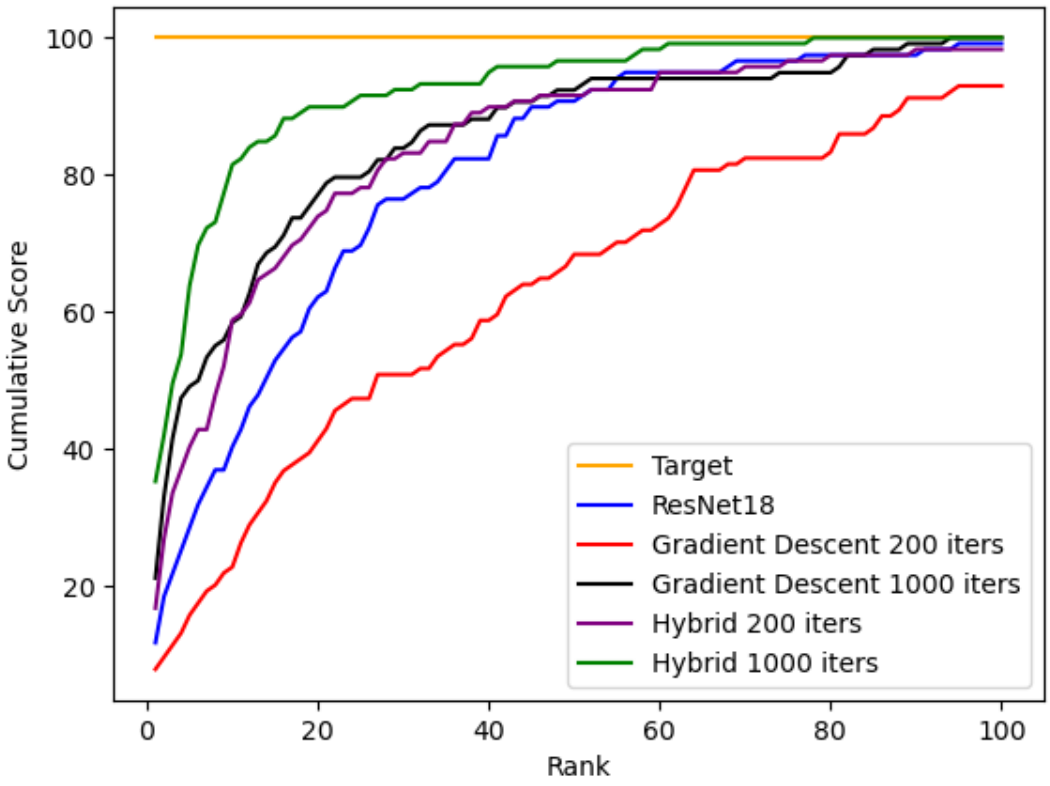

Figure 2.16: Comparison between different inverse mapping methods based on face recognition performance of reconstructed faces.

the target. First, we selected 200 iterations for the hybrid method, but as you can see in column (c), results are blurry however, compare to the ResNet mapping (column (b)), the output looks more similar to the original target. Additionally, the hybrid method with 200 iterations is much better than gradient descent with 200 iterations (column (d)). Hence, we were motivated to increase the number of iterations to 1000 to see if results further improve. It can be observed in column (e) that increasing the number of iterations leads to higher quality reconstruction of natural faces. In order to have a fair comparison between the ResNet18 mapping, the hybrid method and gradient descent approach, we reconstructed AR face images with sole gradient descent for 1000 iterations as well (see column (f)). While some faces can be reconstructed with 1000 iterations of gradient descent, most of them have major artifacts and require many more iterations to be acceptable.

2.3.4 Experiment4: Evaluation based on face recognition performance

One metric to evaluate the quality of reconstructed faces is by comparing their performance in the face recognition task. If a face is recognized better, then it has richer identity information. The images of the AR data set were captured in different illumination conditions,

Table 2.2: Comparison between different approaches based on face recognition performance of reconstructed faces

	Rank1	Rank5	Rank10	Rank20
Target	100%	100%	100%	100%
ResNet18	11.7%	28.5%	40.3%	62.1%
Gradient descent (200 iterations)	7.8%	15.7%	22.8%	41.2%
Hybrid (200 iterations)	16.8%	40.3%	58.8%	73.9%
Gradient descent (1000 iterations)	21.1%	49.1%	58.4%	77.1%
Hybrid (1000 iterations)	**35%**	**63%**	**81%**	**89%**

facial expressions and occlusion. In this experiment, we use the AR data set to compare our ResNet18 encoder and proposed hybrid approach with gradient-based alternatives in terms of face recognition accuracy.

The AR data set consists of 70 men and 50 women (total of 120 subjects). For each, the target image is used as the gallery, and the reconstructed image based on the recovered latent vector is the probe image. The gallery was resized to 128x128 and cropped using MTCNN [52]. Afterwards, each gallery image was projected into latent-space using our ResNet18 inverse mapping model, our hybrid method with 200 and 1000 additional steps and sole gradient descent with 200 and 1000 iterations. We calculated the cumulative match characteristic up to rank-100 between the gallery and the reconstructed probe. First, the facial embedding of all probe and gallery faces is extracted using the VGGFace (ResNet50 architecture) [41] model pre-trained on the vggface2 data set [4]. We used a different face embedding extractor than the FaceNet used for computing the perceptual loss to make sure our results are reliable and valid.

The results of this comparison are shown in Figure 2.16. As depicted in this figure, the hybrid method is noticeably better than sole gradient descent with the same number of iterations. The hybrid method is also better than the ResNet18 encoder solely. A hybrid method with 1000 further steps over the predicted latent vector has the best performance in terms of reconstructed identity information.

Table 2.2 shows a comparison of rank-1, rank-5, rank-10 and rank-20 correct match score between the previously describe inverse mapping models on the AR data set. As demonstrated in this table, the hybrid method outperforms both encoder-based and gradient-based solutions. This indicates that our hybrid method is capable of reconstructing faces that are not only visually appealing, but also contain rich identity information.

2.3.5 Experiment5: Generalizability to other GANs

In this experimen, we evaluate the generalizability of our inverse mapping model to another GAN that the model was not trained on, StyleGAN2. While learning the parameters of ResNet, we used a pre-trained proGAN to generate images based on the predicted latent vectors and computed pixel and perceptual loss. It is expected to see good results for faces synthesize by proGAN. However, we need further investigation to realize whether this approach can be used to map faces synthesized by other GANs to matching latent vectors.

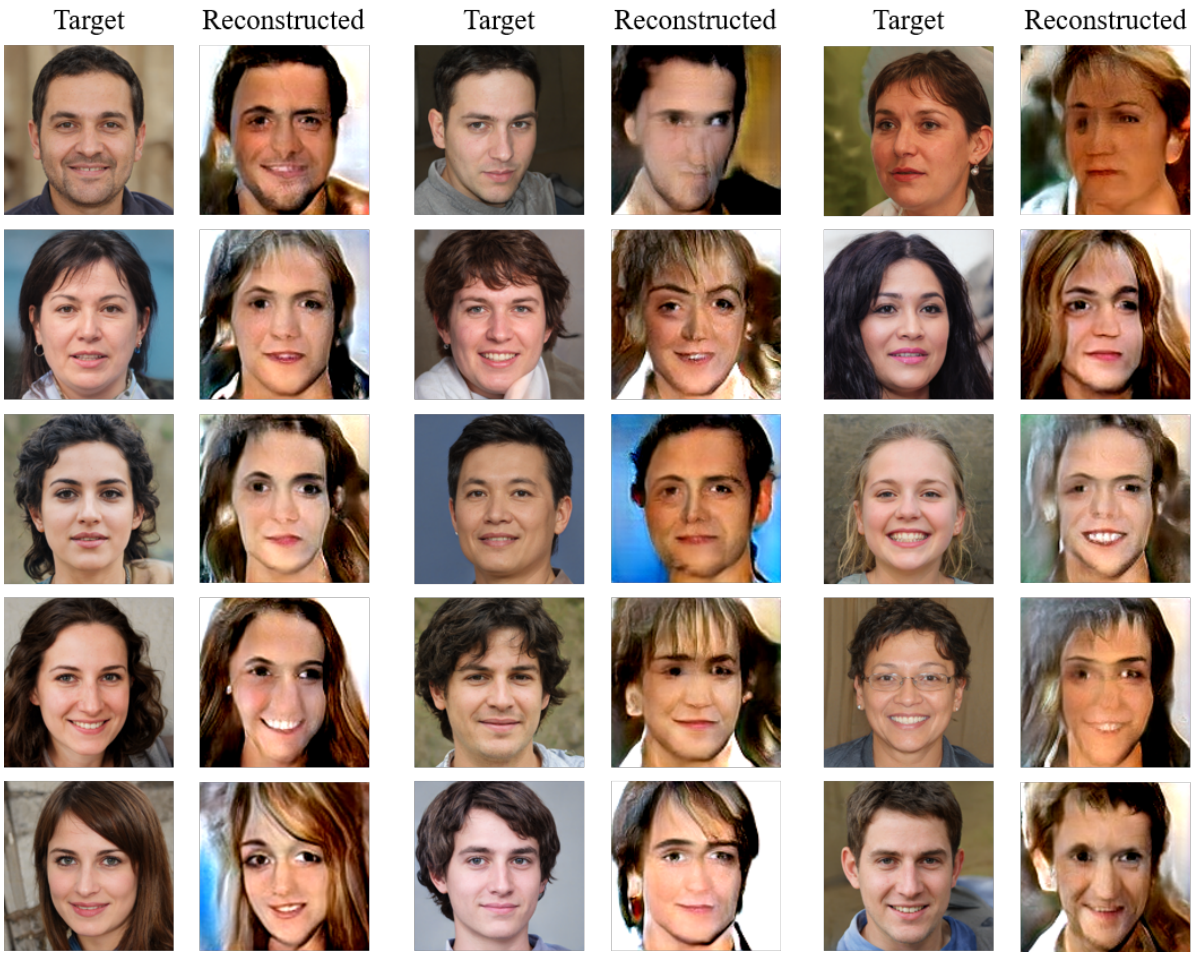

Figure 2.17: Reconstructing StyleGAN2 synthesized faces using predicted latent vectors of our model.

We generated 50 faces with StyleGAN2 generator and align and crop them using MTCNN [52]. Next, we map these images to latent-space using our pre-trained ResNet18 inverse mapper. Figure 2.17 demonstrates the results of reconstructing StyleGAN2 synthesized faces using an inverse mapping model trained with ProGAN. This figure indicates that projecting StyleGAN2 faces to latent space is not as accurate as ProGAN synthesized faces. However, what is interesting in this figure is that even though reconstructions have poor quality, they still contain most of the important facial details of the target. The predicted latent vectors seem to be a good starting point for hybrid methods as well.

2.4 Conclusion

In this chapter, we proposed a method to project generated face images into latent-space extremely fast with high visual quality using deep residual neural networks trained with pixel and perceptual loss. The second major finding was that training the same architecture on pixel and perceptual loss for real faces plus z-loss between generated faces and their ground truth latent

Table 2.3: Comparing gradient-based methods with encoder-based and hybrid approaches based on PSNR, FID score and the computation time (in seconds) for 50 fake faces generated by ProGAN.

Model	**Metric**		
	PSNR	FID	Time (s)
Gradient descent 200 iterations	19.06	30.48	1135.24
ResNet-18 encoder (ours)	16.3	31.22	**3.96**
Hybrid 50 iterations(ours)	**21.96**	**16.94**	32.5467

vectors will bring about a latent vector that contains important facial details and styles of the input face, such as gender, pose, emotion, hair/beard style,etc.

One major limitation of this study is that the ground truth corresponding latent vectors of real face images are not available. This makes mapping real faces much more challenging. We proposed a hybrid method to overcome this issue and reconstruct identical faces to the target. The reconstructed faces using our hybrid method can be recognized by face recognition models with higher accuracy than other inverse mapping alternatives.

The results of this research support the idea that, even though optimization-based approaches are performing well in inverting generators and obtaining accurate latent vectors, deep residual networks in our method are capable of performing the same task on fake faces with relatively identical quality and incredibly faster speed. Table 2.3 summarizes the experiments conducted in this chapter to compare the three most common inverse mapping methods based on an image visual quality metric (PSNR), a reliable GAN evaluation measure (FID) and the time it takes for each approach to map a batch of 50 fake faces into latent vectors.

The insights gained from this study may be of assistance to other problems that involve mapping a huge number of images to their corresponding latent vectors and then finding a new mapping or transformation within the latent-space to generate desired images.

In this study, we substantiated that training a deep neural network on a combination of pixel, perceptual and z-loss can be used to create a latent vector that includes important details and styles of the target real human faces. Though, it still remains an open question if real-time recovery of latent vector for real faces is possible, especially in terms of identity information.

Chapter 3

Inverse mapping of audio GANs

3.1 Overview

Researchers have recently shown an increased interest in mapping generated samples by generative adversarial networks (GANs) into the original latent vectors. GANs are made of a generator and a discriminator network. The generator maps a random latent vector to a generated realistic sample. The discriminator is a binary classifier that aims to determine whether a sample is real or fake. The generator and the discriminator compete with each other to boost their respective performances. After training an audio GAN, for a given random vector z, the generator outputs a realistic audio recording. GAN inversion refers to the task of projecting a given sample (image, audio, video, etc.) to a latent vector that when given to the generator, the output is as similar as possible to the target. Learning such backwards mapping is useful in a wide range of applications.

Projecting generated samples back to the latent-space is a classic problem in computer vision, where the goal is to recover the latent vector of given images. However, investigating the inverse mapping of GANs is still a continuing concern within the audio domain. While methods introduced in the vision domain can be applied to audio as well, they need to be improved and fine-tuned to this domain in order to produce ideal results. Previous works introduced to map images to latent space (z-space) are introduced in the first chapter of this thesis. The second chapter explains our fast novel approach for recovering latent vectors of both natural and synthesized human faces.

In this chapter, we adjust our deep residual neural network (ResNet18) framework to predict latent vectors of audio samples. Our inverse mapping model is trained based on a combination of z-loss and a novel perceptual loss for the audio domain. Later, we demonstrate our results on latent vector recovery of both generated and real audio samples and compare them with sole gradient descent optimization and hybrid methods. Hybrid techniques apply a few steps of gradient descent after the initial prediction of latent representation by the deep neural network. To the best of our knowledge, the experimental work presented here provides one of the first investigations into how to project audio samples to latent representations that can regenerate them. We propose a fast deep network technique to recover latent vectors of both fake and real audio using a novel perceptual loss within the audio domain. We also perform latent vector recovery using gradient descent methods in order to compare this approach with our proposed

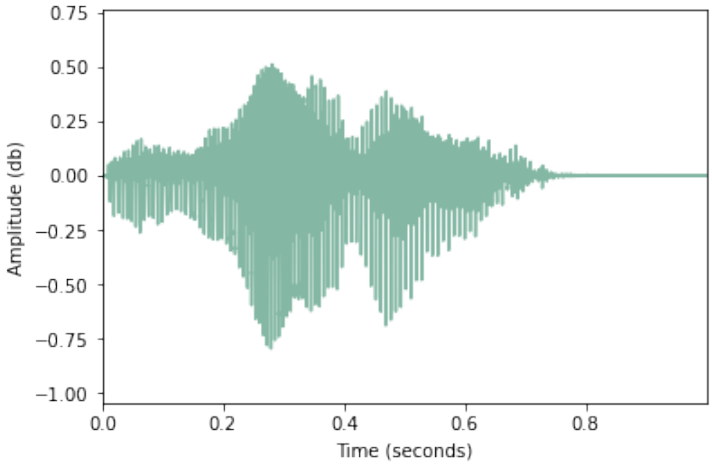

Figure 3.1: An example of a 1-second audio waveform which is sampled with 16 KHz rate.

method. Moreover, we implement a hybrid method that applies a few steps of gradient descent after predicting the representation by the deep network and compare the results with the proposed ResNet-18 framework and sole gradient descent approach.

3.1.1 Audio

The human ear is a sensory organ receiving audio signals, but hearing is a complex computational procedure performed by the brain. The world of computer programming only recognizes numbers. In order to help the computer understand audio, we need to somehow represent it with numbers. Digital signal processing converts audio signals into digital formats (0 and 1) and then mathematically manipulates them. The most significant difference between audio and image is that audio changes over time. Hence, audio signals are formatted as time series.

In order to better understand audio signals, I will explain the most important attributes of an audio signal. Later, different methods of visualizing audio will be described. The frequency is the number of cycles per second. The sampling rate, also referred to as the sampling frequency, is the number of audio samples per second. For example, if a 1 second audio clip has a sampling rate of 16 kHz, it means that the corresponding digital vector of this audio clip has 16000 values. The bandwidth of an audio is the range defined by its maximum and minimum frequency value. If a signal is periodic such as the sine wave, the period of the signal is the length of one complete cycle. The phase of an audio signal is where the cycle begins. If there is a phase shift, it means the beginning of the cycle starts at a different time point.

Waveform

Audio signals can be represented in 2D waveform graphs with time on the x-axis and level (amplitude) on the y-axis. Audio waveform is a way of visualizing the pattern of sound over time. Figure 3.1 shows the waveform of a spoken digit selected from the SC09 data set [47].

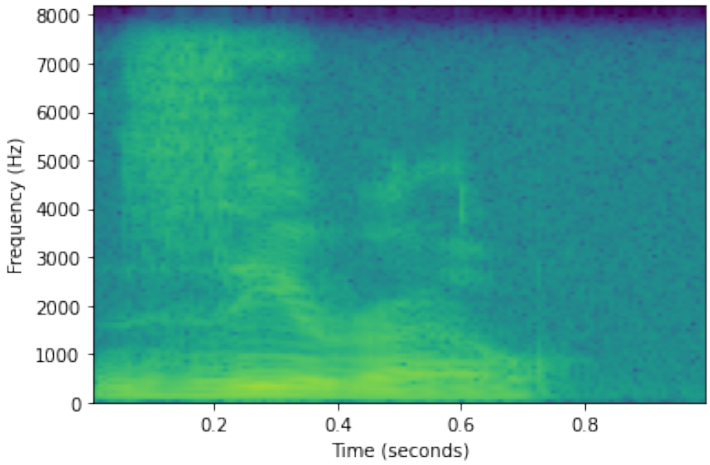

Figure 3.2: An example Spectrogram

Spectrogram

A spectrogram is a 2-D visualization of the spectrum of frequencies of time-series (for example an audio signal). It is possible to convert an audio to spectrogram and then convert the spectrogram back to an audio. This attribute is extremely helpful in machine learning because it makes applying computer vision models on audio data possible. The x-axis of the plot represents time and from left to right, time increases. The y-axis indicates frequency. In spectrograms, the amplitude of the audio frequency at a given time is represented by the third dimension, color. The corresponding spectrogram of the waveform depicted in Figure 3.1 is shown in Figure 3.2. In this figure, lighter colors illustrate a higher amplitude.

3.1.2 Fake audio generators

In this section, the most common audio generators and the available pre-trained models of each generator will be described. Later, we use the outcome of this investigation to choose a generator for our inverse mapping model. An audio signal is a time series that represents the dynamics of the corresponding sound wave over time. The amplitude at each time point can be obtained from the audio waveform function. Since this function is continuous we need to select a sampling rate to achieve a finite set of numbers per audio. Audio signals can also be encoded into 2-dimensional representations by converting them into spectrograms.

WaveGAN

WaveGAN was the first attempt at applying GANs to unsupervised synthesis of raw-waveform audio [7] introduced in 2018. The model has the same architecture as DCGAN [42] with the same number of parameters and operations. WaveGAN is capable of producing one second clips at a frequency of 16 kHz learned from a variety of audio data sets. When the sampling rate is 16000 Hz, it means we get the amplitude of the audio once every 1/16000 seconds. The transposed convolution technique was originally used to up-sample low-level latent vectors to high-level images. WaveGAN makes use of transpose convolutions to up-sample from a latent

space to the final raw-waveform audio using one-dimension filters of length 25. If the sampling rate is equal to 16 kHz, WaveGAN has to generate a vector of length 16000 that corresponds to one second of audio. Each up-sampling block up-samples the input vector by a factor of 4 (instead of 2 in DCGAN) because the initial random latent vector is of size 100 and the output of the generator must be of length 16000. If the scale factor was two, then many layers were required to achieve the desired 16000 length. The output vector should be a power of 4 which means for generating 1 second audio clips, we need to have vectors with a size of at least 16000, the closest power of 4 number to 16000 is 16384 which is the size of the output of WaveGAN.

To avoid checkerboard artifacts in audio, a phase shuffle method is applied to perturb the phase of the discriminator's activations by -n to n. Before applying the phase shuffle, the discriminator could easily detect the fake audio based on the artifacts which would ruin the training process. The difference between WaveGAN and DCGAN include the use of a one dimensional filter of size 25 instead of 5x5 filters (because the audio signal is 1D), increasing the stride from 2x2 to 4 and removing the batch normalization from both generator and discriminator. WaveGAN was trained on human speech, birds sounds, drums and music. The most challenging data set is human speech, similar to the human face being the most challenging domain in Computer Vision. One reason for that is the fact that we might not easily detect a fake bird sound, but we humans have a deep understanding of human voice and speech which makes detecting a fake spoken word much easier than a fake music. Even though the generated audio by WaveGAN seem promising, there is still a large space for improvement.

We chose to use the official WaveGAN model pre-trained on the SC09 (subset of Speech Commands) data set for two reasons [47]. First from a qualitative perspective, it is much easier to distinguish whether the style and content of a reproduced spoken digit remained intact rather than drums or bird sounds. Second, the perceptual loss can be taken from feature maps of a pre-trained classifier on the SC09 data set. There are pre-trained WaveGAN models on SC09 spoken digits data set [47], Drum sound effects and Bach piano performances.

WaveNet

WaveNet [39], invented by DeepMind, combines deep learning techniques from Computer Vision with audio processing methods in order to generate raw audio. WaveNet is used by Google to convert their digital assistant sound to a more similar to human voice audio. WaveNet is trained for the text-to-speech (TTS) task. Parametric TTS was introduced to promote building deep models that contain all the information about the speech synthesis process in their parameters. With parametric TTS, the synthesized audio can be altered by modifying the input of the model, for example changing the speaker ID in order to alter the voice. WaveNet directly models the raw waveform of the audio, one sample at a time. This results in generating more realistic audio. The one dimensional WaveNet is inspired by the two dimensional pixelNet [40][45].

WaveNet is a fully convolutional neural network where the receptive field of each layer grows exponentially to allow for the generation of the final audio. The network is trained on real human voices. After the network is trained, each value of the sampling rate is obtained based on a probability function computed by WaveNet. Afterwards, this value is appended to the network and then the updated input will be used to predict the next value. The intuition here

is that adjacent frames in an audio are dependent on each other and therefore each value should be predicted based on the previous values. This process takes a lot of time and requires too many computations, however, it seems to be essential for generating realistic audio. The text is first converted into a sequence of linguistic and phonetic features, then fed into WaveNet to generate speech. WaveNet was trained using Google's TTS data sets and outperforms Google's TTS models. It has also achieved a high Mean Opinion Score (MOS) by humans. The network is conditioned by the identity of the speaker. This makes it possible to change the style (gender, age, tone, etc.) of the synthesized audio. Even though WaveNet is one of the best audio synthesis tools available, it does not convert a random latent vector to audio, rather it receives a text with a speaker ID as the input. Hence we did not choose this network to train our inverse mapping model. There are WaveNets pre-trained on LJSpeech and CMU ARCTIC data sets available.

GANSynth

GANSynth [11] is another GAN-based model for generating high-quality audio that was published in 2019. Audio waveforms are usually periodic, in addition, human perception is highly sensitive to irregularities in waveforms, this means the task of generating audio by GANs is much more challenging than images. An audio generator must know the proper combination of frequency and phase at each point of time to generate a coherent and realistic sound. On top of that, generating audio with GANs is much more difficult than auto-regressive models such as WaveNet. However, pre-trained GAN generators are capable of generating real-time fake audio. GANSynth generates the entire sequence at the same time. While this is around 50,000 times faster than WaveNet, it has lower fidelity. GANSynth is trained on the NSynth data set of musical records [12]. This work assays the progressive growing GAN architecture in the vision domain for the audio generation task.

They demonstrate the superiority of GANs with sufficient frequency resolution in the spectral domain in generating natural-sounding audio over WaveNet based on both automated metrics and human opinion. Nevertheless, this GAN is only pre-trained on NSynth data set of musical notes. For the inverse mapping task, if we choose musical sounds rather than human spoken digits, it is very difficult to know how similar is the reconstructed audio to the target. Thus we opted to train our inverse mapper based on WaveGAN pre-trained on SC09.

3.1.3 Audio Data sets

In this section, we will describe common categories of available audio data sets and the most famous collections that belong to each group. There are available audio data sets of speech, spoken word, spoken digit, drums (a collection of single drum hits recorded from drum machines), piano, bird sounds and so forth.

The speech commands zero through nine (SC09)

The speech commands zero through nine (SC09) audio data set [47] includes spoken English digits (zero to nine) by many speakers collected using an open-source web application. Each audio sample is one second but the spoken digit is not aligned within this time period. Each

digit is repeated 1850 times in the training set. This data set in total is equal to 5.3 hours of speech. All of the samples were captured using phone or laptop microphones to be in accordance with real-life scenarios. All speakers were in a room with a closed door when recording their voices. The data set also includes 10 commonly used words in robotics which are the following: ”Yes”, ”No”, ”Up”, ”Down”, ”Left”, ”Right”, ”On”, ”Off”, ”Stop”, and ”Go”. The second version of the data set added a few more words. In order to control the quality of recorded audio, they conducted an experiment and asked human workers to reject any audio that the spoken digit is not detectable.

Speech

The LJ Speech Data set[1] is a public data set that contains 13100 audio recordings. There is only one speaker in this database who is reading passages of non-fiction books published between 1884 and 1964. Audio clips can be from 1 to 10 seconds long. The overall data set is 24 hours of speech.

The CMU_ARCTIC databases [27] were designed and collected by the Language Technologies Institute lab at Carnegie Mellon University (CMU). All the audio recordings are recorded by one American speaker (US English) with the main purpose of speech synthesis. This database contains 1150 audio recordings of out-of-copyright texts. The sampling rate is 16 kHz and the audio samples are fully labeled. The recordings are free of noise. They chose words that are easily read by a native English speaker.

NSynth

NSynth [12] is another audio data set that contains 305,979 notes from 1,006 different musical instruments with various pitches, timbres, and volumes. This data set is very much similar to the CelebA [32] for images, all audio samples are aligned and recorded in isolation to reduce variance. NSynth was created to become the baseline for audio machine learning projects. Each audio in this data set has an array of attributes which is beneficial for training conditional GANs. These attributes include the name of the instrument, pitch, velocity (25, 50, 75, 100, 127), note, sample rate, instrument family, instrument source (acoustic, electronic or synthetic) and acoustic quality. Each audio sample is 4 seconds with a sampling rate of 16 kHz.

Bird vocalizations

There are a few audio data sets that contain bird sounds. These data sets are usually used for bird audio detection problems that aim to decide whether there is a bird sound in the given audio or not. Freefield 1010 project [44] is an open audio data set of 7690 audio samples that are collected from the field-recording label in the Freesound audio database[2]. The audio clips are 10 seconds or longer. This data set includes audio from a diverse range of locations and environments. There is another bird sound database called ten-second smartphone audio recordings. This collection of 8000 audio recordings is extracted from a bird sound crowd

[1] https://keithito.com/LJ-Speech-Dataset/, visited 2020-11-1

[2] http://freesound.org, visited 2020-10-31

sourcing research project named Warblr[3] and has recordings from all around the UK. The audio clips include weather and traffic noise as well as human bird imitations. There is another research project called TREE[4] that investigates the Chernobyl Exclusion Zone. The remote monitoring equipment installed at the location have recorded many bird vocalizations. Additionally, there are remote monitoring systems located at Ithaca, NY, USA. In fall 2015, the BirdVox project[5] collected 20000 audio recordings using this equipment.

Large vocabulary speech (TIMIT)

TIMIT [13] is a data set collected in 1993 for the automatic speech recognition task with the purpose of providing acoustic-phonetic researches with speech data. All the recordings are in American English recorded by 630 different speakers. The sampling rate of each waveform is 16 kHz. The speakers were asked to read 10 phonetically rich sentences. The recordings were collected at the Massachusetts Institute of Technology (MIT) and the National Institute of Standards and Technology (NIST) produced the corresponding CD-ROM version and were later verified by human workers.

AudioSet

There are some labeled audio data sets that contain numerous classes of audio clips. AudioSet [14] includes 527 audio classes with around 2M hand-labeled audio recordings. Each sample is 10 seconds long and is collected from YouTube videos. This database overall includes 5.8 hours of continuous audio. Nevertheless, such data sets are too complex for the inverse mapping problem and might mislead the training procedure.

3.2 Methodology

A number of techniques have been developed for recovering latent vectors of images. Including optimization-based methods, auto-encoder inspired architectures, and deep neural network models. A major advantage of using deep residual neural networks is that they are incredibly fast when compared to gradient-based alternatives. The second advantage of using the ResNet architecture is that we can train it on a data set of generated audio and their corresponding ground truth latent vector to learn the inverse mapping. Here, the ResNet-18 architecture previously proposed for the latent vector recovery of images in the second chapter is assayed for the audio domain using generated audio and matching latent representation as well as a novel perceptual loss in this domain. We also use an audio generator network, WaveGAN pre-trained on spoken digits from the SC09 data set, instead of ProGAN.

3.2.1 Inverse Mapping Model

Our goal for the inverse mapping model was to take as input the spectrogram of generated and real audio and output the predicted latent vector for a pre-trained WaveGAN. The reason

[3] http://warblr.net, visited 2020-10-31

[4] https://tree.ceh.ac.uk/, visited 2020-10-31

[5] https://wp.nyu.edu/birdvox/, visited 2020-10-31

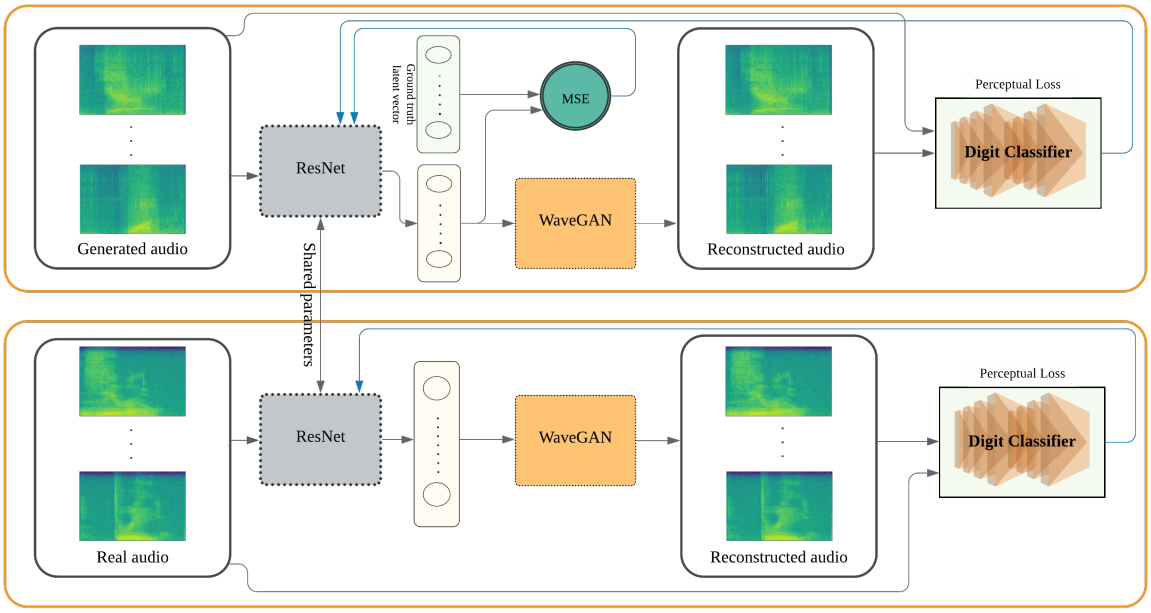

Figure 3.3: The Two Branch Architecture of Our Deep Network Based Inversion Technique for Real Audio.

we chose WaveGAN as the pre-trained audio GAN is that there is a pre-trained WaveGAN model on SC09 audio database of spoken digits from zero to nine. Detecting whether the recovered latent vector is capable of regenerating the target digit is much easier than music or bird sounds. The inverse mapping model architecture is the same as ResNet-18 with the last activation function removed and the number of classes set to the size of the WaveGAN latent space which is 100. The ResNet-18 architecture was selected as it has been successful in latent vector recovery of face GANs.

In the case of synthesized audio, two losses were used to train the network. The first was the MSE between the original latent vector used as input to the WaveGAN and the predicted latent vector by the inverse mapping model (z-loss). The second loss term is the perceptual loss of the audio synthesized by the original latent vector and the reconstructed audio from the predicted latent vector calculated through a classifier model trained on predicting the spoken digit. The idea of computing audio perceptual loss using this classifier is novel and has not been used within this domain before.

To map real audio to latent-space, we cannot use the z-loss as we do not have a true latent vector. The z-loss between synthesized audio and their reconstructed peers in combination with the perceptual loss between real and reconstructed audio helped the model to learn the inverse mapping from both real and synthesized audio to the latent space while maintaining content and style. However, if we only train the ResNet on the perceptual loss between the real audio and its reconstruction, the model forgets the mapping from audio to latent space. In order to tackle this problem, at each epoch the inverse mapping model was trained one batch on z-loss between audio synthesized by WaveGAN from a random latent vector and audio generated by the predicted vector using ResNet18. In the next batch, the perceptual loss between real

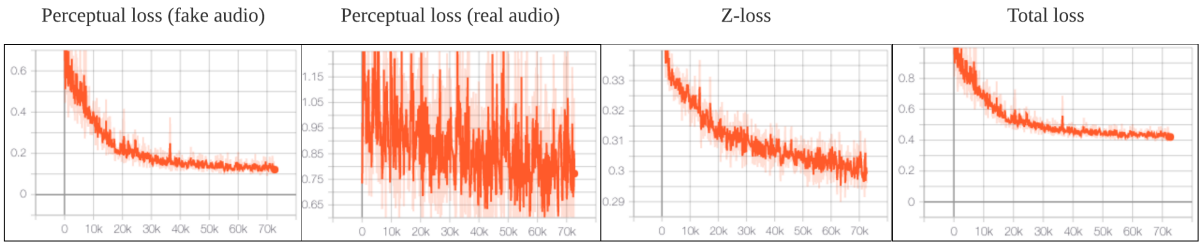

Figure 3.4: The decrease of different loss terms of the objective function over time.

audio from the SC09 data set and reconstructed counterpart using the ResNet18 model were fed backwards to the model to fine tune the inverse mapping for the real audio domain. The architecture of our proposed method is presented in Figure 3.3.

Objective Function

To train the inverse mapping model, our objective function includes two loss terms:

- **MSE between latent vectors (z-loss):** At run time, audio was synthesized by WaveGAN from a random latent vector with a uniform distribution between -1 and 1. With the synthesized audio as input to the inverse mapping model, the latent vector was predicted in the last layer of the network. The ground truth latent vector was then compared to the predicted latent vector through Mean Squared Error (MSE). This loss is necessary to learn the inverse mapping between the audio and latent vectors.

- **Perceptual loss:** For real and synthesized audio, the perceptual loss was back-propagated through the inverse mapping model. The perceptual loss was quantified as the difference in activations in a ResNet-18 spoken digit classifier's output at each residual block. The classifier was pre-trained on the SC09 data set with 95.41% accuracy. This loss improved the reconstruction of real audio and helped keep style and content intact.

The decrease of loss when training ResNet model is depicted in Figure 3.4. Even though there is a lot of variation in the perceptual loss of real audio, the overall loss function is decreasing.

3.2.2 Implementation Details

The inverse mapping model was trained for 250 epochs with an Adam optimizer with the learning rate of 0.001 and a batch size of 64. The only modifications to the ResNet-18 architecture was the removal of the final activation function and the number of classes being set to the size of WaveGAN's latent space (100). During each epoch, the model was first trained on one batch of real audio with perceptual loss and then one batch of WaveGAN synthesized audio with MSE between latent vectors and perceptual loss between the original audio and reconstructions.

The code for audio inverse mapping is written in Tensorflow 2 framework. WaveGAN models are stored in Tensorflow 1 format but are converted at run-time to be compatible with

Table 3.1: Synthesized Audio Reconstructions

	Inception Score (mean ± std)	FID	MSE (raw audio)	SSIM (spectrogram)
Fake	7.23 ± 0.003	-	-	-
Gradient-based	3.96 ± 0.203	0.7140	0.00489	0.9618
Inverse Mapping Model (ours)	7.50 ± 0.288	0.4841	0.00196	0.9731
Hybrid Method (ours)	7.23 ± 0.006	0.4850	0.00003	0.9979

Tensorflow 2. In order to generate fake audio we use WaveGAN pre-trained on the SC09. WaveGAN code and pre-trained models are publicly available. The WaveGAN model is frozen in order to run in Tensorflow 2 at run time.

3.3 Experiments and results

In this section, we compare our ResNet-18 inverse mapping model with sole gradient descent and hybrid methods based on MSE distance, SSIM measure, digit classification accuracy, Inception Score (IS) and Frechet Inception Distance (FID) score for both synthesized and real audio.

3.3.1 Data set

Our generated audio data set was created at run time by generating a random latent vector of size 100 with a uniform distribution between -1 and 1. The latent vector was used as input to WaveGAN to generate raw-waveform audio. For real audio, we used the SC09 data set which contains spoken digits from zero to nine. This data set is described in the audio data sets section. The training set includes 18620 samples with roughly equal occurrences of each digit by many different speakers.

3.3.2 Experiment 1: Mapping synthesized audio to latent space

An initial objective of this project was to map generated audio back to the corresponding latent space. We trained a ResNet18 deep network to perform this task inspired by our result in the second chapter on face images. The purpose of Experiment 1 is to evaluate the performance of our inverse mapping model trained using z-loss and perceptual loss in the latent vector recovery of synthesized audio. Furthermore, we compare the sole ResNet18 model with gradient descent and hybrid alternative methods. The results in Table 3.1 show that we were able to recover the latent vector of synthesized audio with high accuracy. While hybrid techniques seem promising in the latent vector recovery of images, they are not as helpful in the audio domain. Even though the MSE and SSIM metric demonstrate the superiority of the hybrid method, in actuality, the reconstruction through our sole ResNet18 inverse mapping model produces a more coherent spoken digit with better content and style reconstruction. This can

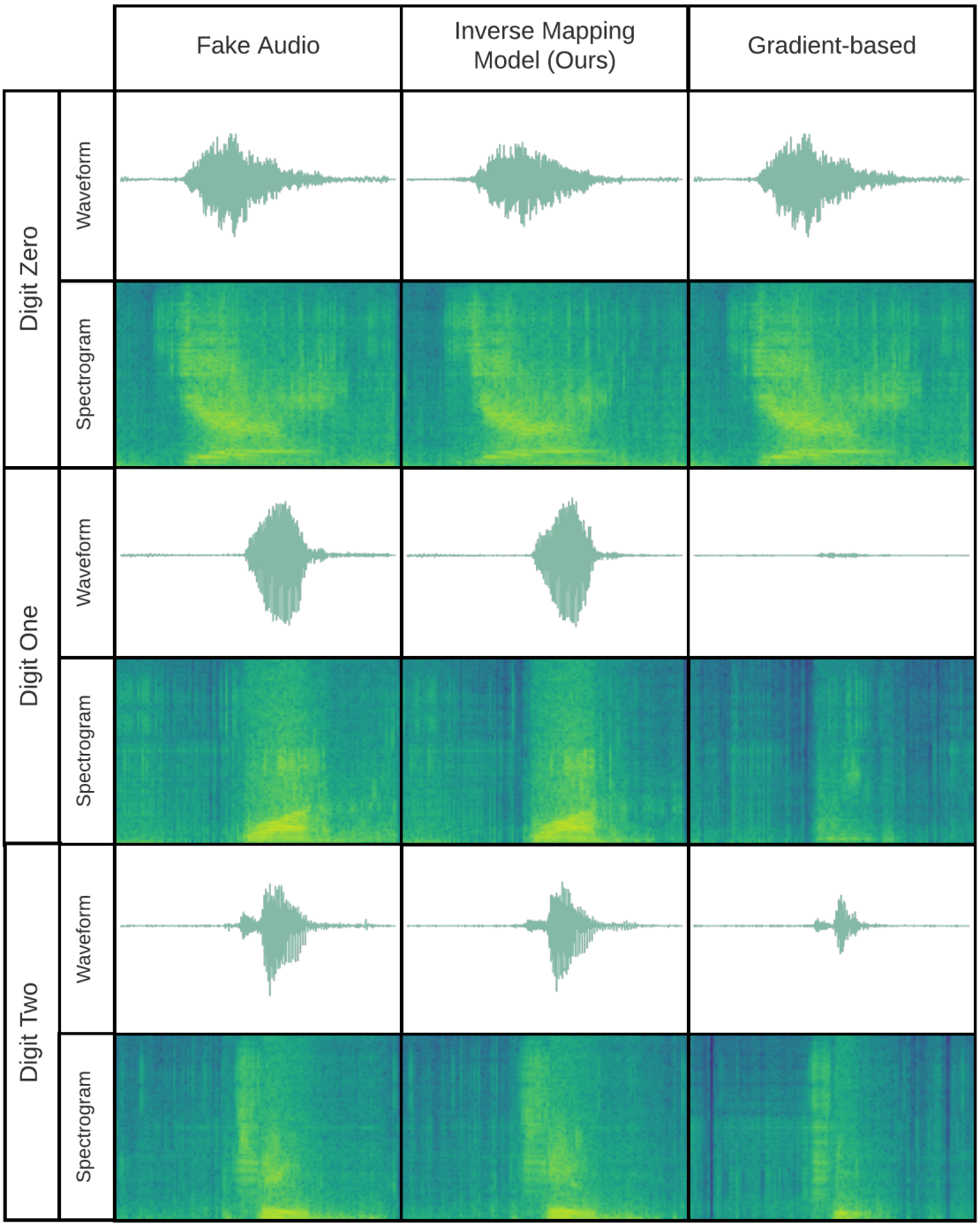

Figure 3.5: Comparing the Performance of Deep Network Inverse Mapping Model with Gradient Descent Approach in Latent Vector Recovery of Fake Audio. Three Example Digits are Shown Here.

be shown through the inception and FID score of our inverse mapping model. It is further verified in Experiment 3 where reconstructed audio clips obtained reasonable accuracy in the digit classification task. The qualitative results of our approach on fake audio are shown in Figure 3.5. In this figure, we showed three sample digits of zero, one and two. The spectrogram and waveforms of ResNet18 reconstructions are more similar to the target. The reconstructions using the gradient descent approach were usually so quiet and difficult to hear, this can easily be observed in their waveforms.

3.3.3 Experiment 2: Comparing Different Approaches

As mentioned earlier, optimization methods are capable of updating a random latent vector to recover the desired latent vector that reconstructs the target audio. We implemented an optimization-based method that updates a random vector for a maximum of 50000 steps using gradient descent. Finding the perfect learning rate for gradient descent is extremely challenging. In addition, we developed a hybrid method that first predicts a latent vector using our inverse mapping model and then updates it further with a maximum of 200 steps of gradient descent. The comparison between these three methods is summarized in Table 3.1 for fake audio. Although the hybrid method produces a lower loss for the reconstruction of raw-waveform and spectrograms, the reproduced audio is of worse auditory quality. The audio related to this project is included in the supplementary materials of this thesis where you can compare the audio reconstructed using different methods. This indicates the inverse mapping model captures the important style and content of the audio while discarding noise. Overall, these results indicate that our inverse mapping model performs better than optimization-based methods. The comparison between the mentioned approaches on real audio is summarized in Table 3.2.

Gradient Descent

In order to recover the latent representation of a given audio using this method, we first sample a random vector, z^*, of size 100 from the uniform distribution. Then we feed this latent vector to WaveGAN to get the corresponding synthesized audio. Next, we convert this audio to a spectrogram and compute the Mean Absolute Error (MAE) loss between this spectrogram and the spectrogram of the target audio. Afterward, z^* is updated by the SciPy optimization library using the L-BFGS algorithm [31]. After each iteration, the updated z^* becomes closer to the ground truth latent vector. Equation 3.1 depicts the optimization process. $\mathcal{W}$ stands for pre-trained WaveGAN and $\mathcal{S}$ converts audio to spectrogram. The main disadvantage of the optimization methods is that they are very slow. Numerous iterations are required to find the optimum latent vector for a target sample; further, finding the perfect hyper-parameters (learning rate and number of iterations) is really difficult. To recover latent vectors of real audio, we limited z^* to 50000 gradient steps.

$$min_{z^*} = \frac{1}{n} \sum \|\mathcal{S}(\mathcal{W}(z)) - \mathcal{S}(\mathcal{W}(z^*))\|_1 \tag{3.1}$$

Hybrid

The hybrid method combines gradient descent optimization with deep neural network predictions. One problem with gradient descent is that the initial latent vector is very important in obtaining good results. Hybrid methods first use a pre-trained deep network to predict an initial latent vector. Next, they further improve it via gradient descent. This not only solves the initial point problem but also decreases the number of required iterations. In the second chapter of this thesis, the ResNet18 inverse mapping model for face images was further improved through a few steps of gradient descent. The improvement was demonstrated both quantitatively and qualitatively. In the vision domain, there was a direct relationship between the number of optimization iterations and the quality of the reconstructed face. In other words, as the number of iteration increased, more precise latent vectors were recovered. We investigate the hybrid method in the audio domain with a various number of iterations and learning rates. Nevertheless, it seems that the hybrid method is not as helpful in the audio domain.

Table 3.2: Real Audio Reconstructions

	Inception Score (mean ± std)	FID	MSE (raw audio)	SSIM (spectrogram)	Accuracy
Real	9.20 ± 0.019	-	- -		95.41%
Gradient-based	2.20 ± 0.063	1.0566	0.00959	0.953	17.17%
Encoder-based (ours)	7.93 ± 0.077	0.4514	0.01176	0.946	71.06%
Hybrid (ours)	3.89 ± 0.027	0.8976	0.00887	0.954	36.93%

3.3.4 Experiment 3: Real Audio Reconstruction

Very little was found in the literature on how to reconstruct real audio using GAN inversion techniques. A major problem with mapping real audio is that there are no ground truth latent vectors that accurately generate them. This means that we cannot compute the z-loss for them. Another issue is the limitation of the audio GANs themselves, they are not capable of generating high-quality naturalist audio that sound identical to the real recordings, such as spoken digit audio recorded by humans. Nevertheless, our proposed approach is universal and can later be applied to invert more advanced GANs. The results of comparing our inverse mapping model with the alternatives on projecting real audio to latent representations are summarized in Table 3.2. The accuracy column represents the accuracy of the reconstructed audio in the digit classification task. If a digit is classified correctly, it means the reconstruction is semantically correct. What stands out in the table is that while gradient-based methods reconstruct the waveform and spectrograms with good quality, in terms of classification accuracy, our inverse mapping model is significantly better and is closest to the real audio accuracy. Together these results provide important insights into the latent vector recovery in the audio domain. The comparison between different inverse mapping methods on real audio is presented in Figure 3.6. Three example digits of zero, one and two have been selected.

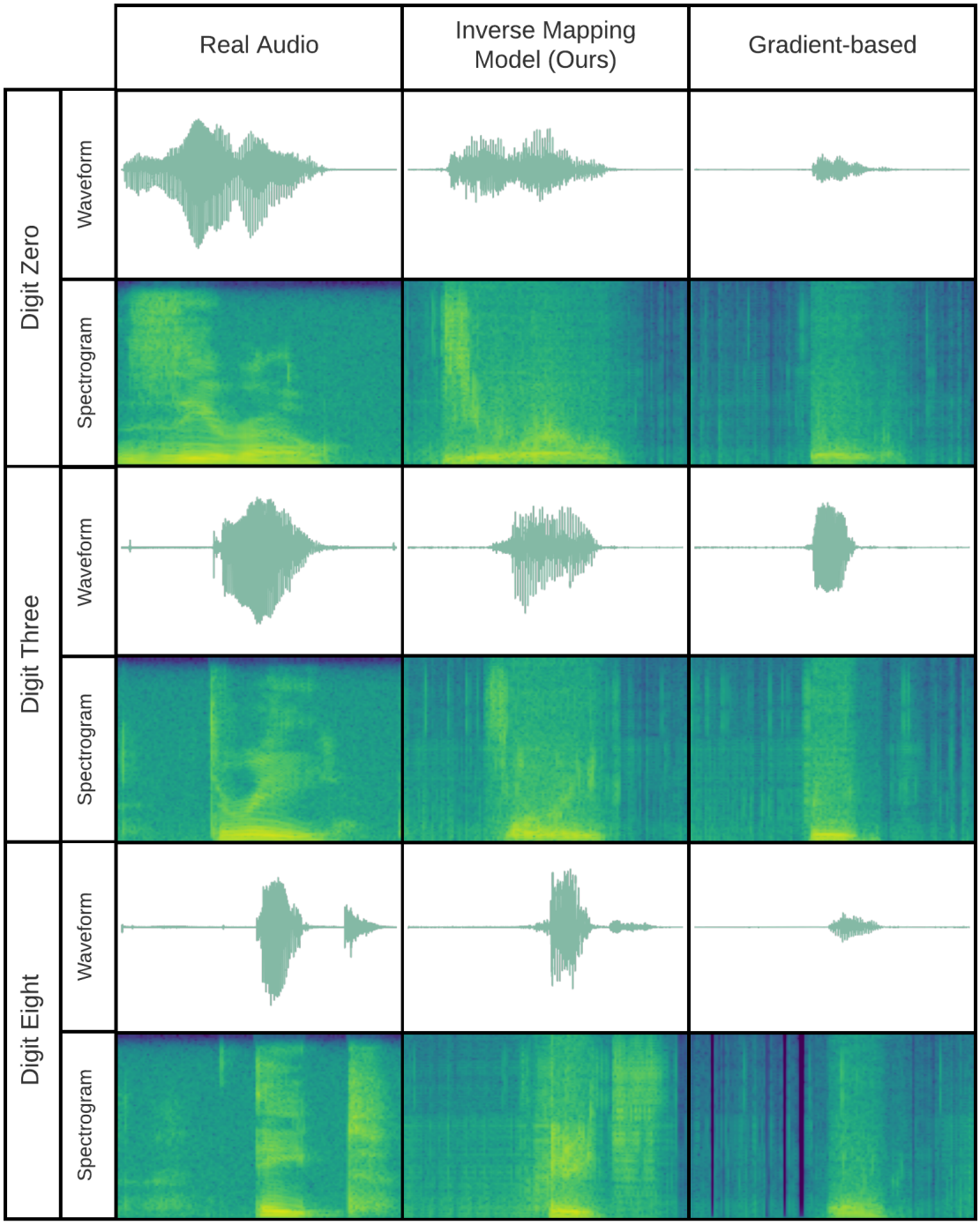

Figure 3.6: Comparing the Performance of Deep Network Inverse Mapping Model with Gradient Descent Approach in Latent Vector Recovery of Real Audio.

3.4 Conclusion

This chapter aimed to examine different approaches of inverse mapping audio GANs. We proposed a deep residual neural network based approach to project both synthesized and real audio into latent space. We trained a ResNet-18 architecture based on a combination of z-loss and perceptual loss. This study has identified that deep network solutions are not only faster than optimization-based alternatives, but also more accurate in terms of both auditory quality of the reconstructed sounds and digit classification accuracy.

Taken together, these findings suggest a role for deep neural networks in promoting the inverse mapping of audio GANs. Being limited to current audio GANs, this study lacks high quality reconstructed audio for real sounds. However, our method is universal and can later be applied to more advanced GANs.

Chapter 4

Discussions and Conclusion

4.1 Conclusion

The present thesis aimed to examine the possibility of training a fast inverse mapping from a human face or voice into a GAN's latent space. This research was undertaken to design a latent vector recovery model for both images and audio. Using the recovered latent vectors, we aim to reconstruct the target input. We call this an inverse mapping problem because the original pipeline of GANs receives a random latent vector input and generates an image/audio. In this research, we plan to go backwards, by receiving an image/audio we want to predict a corresponding latent vector that reconstructs the input as identical as possible. The second aim of this study was to investigate the possibility of mapping a natural face image or real human voice into latent space. We have found that our method can find matching latent vectors that include most of the important details of the target.

There are three major approaches to the inverse mapping of GANs. The most commonly used method is optimization via gradient descent algorithm. By starting from a random latent vector, the loss between the image/audio generated by this initial vector and the target is fed backwards to update the latent vector. This process is continued for many iterations until the latent vector is good enough or in other terms, the loss is below a threshold. Choosing a good set of hyper-parameters as well as a good starting point is crucial to obtain good results. Moreover, this method is incredibly slow. That's why encoder-based solutions were proposed. These methods train an encoder to learn the mapping between generated examples and latent space. Even though this method adds some parameters to the training and has the risk of over-fitting, it has the big advantage of performing real-time mapping. The previous encoder-based studies did not show promising results in particular when compared to gradient-based alternatives. There is a third category of hybrid methods that first predict a latent vector using an encoder, later they optimize it using a few iterations of gradient descent.

We proposed a deep residual neural network (ResNet18) for this inverse mapping problem. The network is trained on a data set of synthesized samples and their ground truth latent vectors (z-loss). Moreover, the perceptual loss between reconstructed and target examples helped the model include texture details of the target in the reconstructed peer. We proposed two different frameworks for fake and natural human faces because we cannot compute the z-loss on real faces. We also proposed a novel perceptual loss in the audio domain based on a pre-trained

digit classifier.

We evaluated the recovery performance based on quantitative measures such as SSIM, PSNR, Inception Score, FID score, face recognition and digit classification accuracy. The results of our experiments indicate that the hybrid method for images seems to be the best performing approach. However, the hybrid method made no significant difference to sole ResNet18 for audio. On top of that, we compared our method with alternatives qualitatively by comparing target and reconstructed images/spectrograms. Additionally, we compared different approaches based on computational time. Our real-time deep neural network approach is three orders of magnitude faster than optimization based alternatives. This study has found that generally inverse mapping of real image or audio is much more challenging than synthesized samples.

These findings suggest that in general, our method is better than the state of the art especially in terms of computation time. In addition, our method is universal and can be applied to more advanced GANs to get higher quality results at the cost of more training time. The findings of this research provide insights for many classification and retrieval tasks that rely on finding a fast accurate mapping between generated samples by GANs and their matching latent vectors.

4.2 Discussion

This study contributes to the understanding of GAN latent vectors. Even though the problem of latent vector recovery for given images can have multiple solutions, we have shown that our method is capable of finding one of them. The solution is not unique and might vary if we run the test on the same image multiple times, however, the reconstructed result looks similar to the target every time, in particular for synthesized faces. This task is slightly different for audio GANs, there are examples that reconstructed audio clips do not resemble the target. The evidence from this study suggests that this is not a limitation of our inverse mapping model, this is in fact a limitation of the currently available audio GANs. The performance of current audio GANs is inferior compared to image GANs. Not every random latent vector leads to a good quality audio. Therefore, some reconstructions do not sound as expected. We believe that this problem will be solved when more advanced audio GANs are introduced in the future. The findings of this study suggest that mapping natural images and audio to latent space with high speed is possible however the quality is not as good as synthesized samples. There is still room for improving the mapping between natural human faces to latent representations.

In this thesis, a comprehensive review of inverse mapping techniques, state-of-the-art image/audio GANs, their capabilities, design details and also limitations have been presented. Throughout this research, an extensive experimentation on hyper-parameters (optimizer, learning rate, number of epochs, etc.) and network architectures (various ResNet architectures) has been performed to develop the best performing inverse mapping model. Alternative techniques (gradient-based and hybrid methods) have been implemented to demonstrate the superiority of the proposed methods. The proposed methods were compared with competing methods on various qualitative and quantitative visual/auditory metrics.

4.2.1 Limitations

The major limitations of this study are lack of ground truth latent vectors per natural samples and the limitation of current audio GANs available. The scope of this study was limited in terms of the category of images and audio used for training and testing the model. For images, we only tested human faces, however, this is because the human face is the most challenging image to generate. Within the audio domain, we only considered human spoken digits since we believe evaluating the integrity and accuracy of the reconstructed audio is easier using this category.

Lack of ground truth latent vector for natural images and natural audio

A limitation of this study is that we do not have ground truth latent vectors for natural examples. Natural images or audio were not generated by GANs originally. They might contain details that GANs have not learned before. This means that we cannot train our model based on z-loss for natural inputs. Given the fact that z-loss is the main reason ResNet18 learns the accurate mapping, it becomes much more challenging to recover latent vectors of natural samples. Our proposed simultaneous training framework solves this issue to a good extend, but there is still potential for improvement. Using the hybrid method, we further improved the results on faces, but this method is not real-time anymore.

The limitation of audio GANs themselves

Whilst GANs have achieved remarkable performance in face generation, the audio generators are not as good. This mainly lies under the time-dependent nature of audio signals. The amplitude of the audio signal at each time point depends on the previous values, if the order is not respected, then the synthesized audio would not be natural-sounding and coherent. This makes generating the whole audio signal in parallel very challenging. Recent audio GANs such as WaveGAN and GANsynth have shown promising results in the audio synthesis task. However, there is still a need for improvement to generate naturalistic reliable audio that is indistinguishable from real audio in particular when generating audio similar to the human voice. Since the GAN themselves are not perfect and realistic, we cannot expect to recover latent vectors that generate realistic results. The difference is most significant when mapping real human voice to latent space. The predicted latent vector, no matter how accurate it is, will be fed into a generator that is not performing well in generating audio identical to the real counterpart. Hence, the reconstruction of real audio is not of the same quality as the target. Nevertheless, the reconstructed spoken digit is mainly the accurate digit.

4.2.2 Applications

Inverse mapping of GANs has many applications in both industry and academic research. One use case of this technique is to evaluate the performance of GANs. If an image or audio cannot be fully reconstructed, it can also mean that the GAN has not learned all the details that exists in the input.

Furthermore, since linear operations in the latent-space results in a semantically meaningful result in the output of the GAN, inverse mapping models can be used as a means to manipulate

synthesized images and audio to generate desired outputs. This is also useful in style transfer where we want to edit an image or for example modify the voice of an audio. This quality of latent vectors can be very helpful in classification or retrieval tasks.

4.3 Future research

The questions raised by this study are whether it is possible to map synthesized image or audio to accurate latent vectors in real-time as well as whether natural human face and voices can be represented by latent vectors that can generate them using GANs. A natural progression of this work is to analyze whether higher quality samples could be reconstructed if a better GAN was used to train the ResNet model. More broadly, research is also needed to determine whether it is possible to build an inverse mapping that is robust against the resolution of images or the existence of noise in audio recordings.

While currently, it seems that recovery of real samples are more challenging, this finding can be applied to deep fake detection applications. There are a variety of studies focusing on deciding whether an image or audio is synthesized or real. If the latent vector recovered by a synthesized sample is much more accurate than the latent vector predicted by inverse mapping models, then this difference can be helpful in discriminating fake from real examples.

This research has raised many questions in need of further investigation. Is it possible to map natural human faces to precisely accurate latent vectors in real time? Would we be able to exactly project real audio to latent-space if we had an advance audio generator? Can we map other image or audio categories to latent space with the same accuracy?

Bibliography

[1] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a GAN cannot generate. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4502–4511, 2019.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[3] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208, 1995.

[4] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[5] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE transactions on Neural Networks and learning systems*, 30(7):1967–1974, 2018.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.

[7] Chris Donahue, Julian J. McAuley, and Miller S. Puckette. Synthesizing audio with generative adversarial networks. *CoRR*, abs/1802.04208, 2018.

[8] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[9] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, pages 658–666, 2016.

[10] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[11] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. Gansynth: Adversarial neural audio synthesis. *arXiv preprint arXiv:1902.08710*, 2019.

[12] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, and Mohammad Norouzi. Neural audio synthesis of musical notes with wavenet autoencoders. *arXiv:1704.01279*, 2017.

[13] John S Garofolo. Timit acoustic phonetic continuous speech corpus. *Linguistic Data Consortium, 1993*, 1993.

[14] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780. IEEE, 2017.

[15] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5744–5753, 2019.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[17] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on Computer Vision*, pages 87–102. Springer, 2016.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[19] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[20] Minyoung Huh, Richard Zhang, Jun-Yan Zhu, Sylvain Paris, and Aaron Hertzmann. Transforming and projecting images into class-conditional generative networks. *arXiv preprint arXiv:2005.01703*, 2020.

[21] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition workshops*, pages 11–19, 2017.

[22] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on Computer Vision*, pages 694–711. Springer, 2016.

[23] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[24] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[25] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[26] Aditya Khosla, Akhil S Raju, Antonio Torralba, and Aude Oliva. Understanding and predicting image memorability at a large scale. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2390–2398, 2015.

[27] John Kominek, Alan W Black, and Ver Ver. CMU arctic databases for speech synthesis. 2003.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[29] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[30] Zachary C Lipton and Subarna Tripathi. Precise recovery of latent vectors from generative adversarial networks. *arXiv preprint arXiv:1702.04782*, 2017.

[31] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 1989.

[32] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[33] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on Computer Vision*, pages 3730–3738, 2015.

[34] Junyu Luo, Yong Xu, Chenwei Tang, and Jiancheng Lv. Learning inverse mapping by autoencoder based generative adversarial nets. In *International Conference on Neural Information Processing*, pages 207–216. Springer, 2017.

[35] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5188–5196, 2015.

[36] Aleix M Martinez. The ar face database. *CVC Technical Report24*, 1998.

[37] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[38] Aaron Nech and Ira Kemelmacher-Shlizerman. Level playing field for million scale face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7044–7053, 2017.

[39] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[40] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[41] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[42] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[43] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[44] Dan Stowell and Mark D Plumbley. An open dataset for research on audio field recording archives: freefield1010. *arXiv preprint arXiv:1309.5275*, 2013.

[45] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016.

[46] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315, 2017.

[47] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

[48] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011*, pages 529–534. IEEE, 2011.

[49] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

[50] Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 192–199, 2014.

[51] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[52] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

[53] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European conference on Computer Vision*, pages 597–613. Springer, 2016.

Curriculum Vitae

Name: Nicky Bayat

Post-Secondary: Bachelor of Computer Sciences
2014 - 2019
University of Tehran
Tehran, Iran

Experiences: Teaching Assistant
The University of Western Ontario
2019 - present

Machine Learning Researcher Intern
AvocadoCore company
London, ON
Mar. 2020 - Oct. 2020

Teaching Assistant
University of Tehran
Tehran, Iran
Feb. 2019 - Jan. 2019

Publications:

1. Nicky Bayat, Vahid Reza Khazaie, and Yalda Mohsenzadeh. "Inverse mapping of face GANs." arXiv preprint arXiv:2009.05671 (2020).

2. Andrew Keyes*, Nicky Bayat*, Vahid Reza Khazaie, and Yalda Mohsenzadeh. "Latent Vector Recovery of Audio GANs." arXiv preprint arXiv:2010.08534 (2020).

3. Vahid Reza Khazaie, Nicky Bayat, and Yalda Mohsenzadeh. "IPU-Net: Multi Scale Identity-Preserved U-Net for Low Resolution Face Recognition." arXiv preprint arXiv:2010.12249 (2020).